

**KLASYFIKACJA OBRAZÓW RASTROWYCH
Z WYKORZYSTANIEM
SZTUCZNYCH SIECI NEURONOWYCH
I STATYSTYCZNYCH METOD KLASYFIKACJI**

**Tomasz Kubik
Witold Paluszyński
Adam Iwaniak
Przemysław Tymków**

**KLASYFIKACJA OBRAZÓW RASTROWYCH
Z WYKORZYSTANIEM
SZTUCZNYCH SIĘCI NEURONOWYCH
I STATYSTYCZNYCH METOD KLASYFIKACJI**

Opiniodawca

dr hab. inż. Beata Hejmanowska, prof. nadzw.

Redaktor merytoryczny

dr hab. inż. Andrzej Borkowski, prof. nadzw.

Opracowanie redakcyjne

dr Ewa Jaworska

Korekta:

mgr Elżbieta Winiarska-Grabosz
Janina Szydłowska

Łamanie

Witold Paluszyński

Projekt okładki:

Tomasz Kubik
Witold Paluszyński

Obrazy satelitarne na rys. 1.1 — Landsat Inc., dzięki uprzejmości prof. Andrzeja Ciołkosza i prof. Stanisława Lewińskiego z Instytutu Geodezji i Kartografii; obraz na rys. 1.2 — pochodzący z projektu CORINE Land Cover 2000 — za zgodą Głównego Inspektoratu Ochrony Środowiska; zdjęcia lotnicze na rys. 2.2, 4.3, 4.8, 4.11 i 4.15, za zgodą Centralnego Ośrodka Dokumentacji Geodezyjnej i Kartograficznej Głównego Urzędu Geodezji i Kartografii.

Publikacja finansowana przez Ministerstwo Nauki i Szkolnictwa Wyższego
Grant badawczy nr 5 T12E 029 24

Monografie LIX

© Copyright by Uniwersytet Przyrodniczy we Wrocławiu, Wrocław 2008

ISBN 978-83-60574-35-5

ISSN 1898-1151

**WYDAWNICTWO UNIWERSYTETU PRZYRODNICZEGO
WE WROCŁAWIU**

Redaktor Naczelny — prof. dr hab. Andrzej Kotecki
ul. Sopocka 23, 50-344 Wrocław, tel. 071 328-12-77
e-mail: wyd@up.wroc.pl

Nakład 100 + 16 egz. Ark. druk. 5,50
Druk i oprawa: Wydawnictwo Tekst Sp. z o.o.
ul. Kossaka 72, 85-307 Bydgoszcz

Spis treści

Wstęp	7
1. Klasyfikacja obszarów na obrazach satelitarnych i zdjęciach lotniczych	9
1.1. Klasyfikacja <i>per pixel</i> i <i>per parcel</i>	10
1.2. Obrazy Landsat TM	11
1.3. Projekt CORINE	13
1.4. Metody analizy skupień	14
1.4.1. Opracowanie danych	15
1.4.1.1. Budowa histogramów	15
1.4.1.2. Wizualizacja histogramów	15
1.4.1.3. Wstępne wnioski	18
1.4.2. Próby klasyfikacji <i>per pixel</i>	18
1.4.2.1. Przygotowanie danych	18
1.4.2.2. Automatyczna klasyfikacja	18
1.4.3. Wizualizacja	19
1.4.4. Wyniki	20
1.5. Naiwny klasyfikator bayesowski	21
1.5.1. Teoria Bayesa	21
1.5.2. Wielowymiarowa, ciągła przestrzeń cech	22
1.5.3. Naiwny klasyfikator Bayesa	23
1.5.4. Klasyfikacja bayesowska <i>per pixel</i>	24
1.5.5. Eksperymenty i wyniki	25
1.6. Ocena wyników klasyfikacji	28
1.6.1. Pierwiastek błędu średniokwadratowego (<i>RMSE</i>)	28
1.6.2. Progowany błędem procent poprawnej klasyfikacji (<i>TCE</i>)	28
1.6.3. Współczynniki obliczane na podstawie macierzy niezgodności	28
2. Segmentacja obszarów metodą momentów chromatycznych	32
2.1. Podstawy matematyczne	32
2.1.1. Konwersja RGB do CIE XYZ	32
2.1.2. Chromatyczność	32
2.1.3. Dystrybucja chromatyczności oraz jej ślad	32
2.1.4. Momenty chromatyczne	33
2.2. Metoda segmentacji	33
2.3. Eksperymenty i wyniki	34
2.3.1. Przygotowanie zestawu uczącego i estymacja parametrów	35
2.3.2. Wyniki segmentacji obrazu	36

3. Eksperymenty z klasyfikacją tekstur metodą losowych pól Markowa	38
3.1. Podstawy matematyczne	38
3.1.1. Definicje podstawowe	38
3.1.2. Pola losowe Markowa i Gibbsa	39
3.2. Model opisu tekstury	41
3.3. Synteza tekstury	43
3.3.1. Metropolis sampler	43
3.3.2. Gibbs sampler	43
3.4. Segmentacja tekstur	44
3.5. Szczegóły implementacji	46
3.6. Eksperymenty	46
3.6.1. Przygotowanie obrazów referencyjnych	46
3.6.2. Generacja zestawu uczącego dla sieci neuronowej i estymacja parametrów	47
3.6.3. Trenowanie sieci	47
3.6.4. Segmentacja sztucznych tekstur	48
3.6.5. Segmentacja tekstur naturalnych	50
4. Detekcja i klasyfikacja obszarów na zdjęciach lotniczych za pomocą sieci neuronowych	52
4.1. Rozpoznawanie tekstur	53
4.1.1. Wariogram	53
4.1.2. Metoda macierzy GLCM w analizie tekstur	53
4.2. Użycie sieci neuronowych z maską	56
4.2.1. Założenia eksperymentu	56
4.2.2. Eksperymenty	60
4.2.2.1. Ocena jakości klasyfikacji	63
4.2.2.2. Wnioski z eksperymentu	64
4.3. Użycie sieci neuronowych z algorytmem identyfikacji tekstur GLCM	64
4.3.1. Metoda GLCM w identyfikacji budynków na zdjęciach lotniczych	65
4.3.1.1. Wnioski	68
4.3.2. Klasyfikacja obszarów zabudowanych na zdjęciach lotniczych z zastosowaniem metody GLCM	68
4.3.2.1. Wnioski	74
4.4. Porównanie klasyfikacji obszarów za pomocą masek oraz cech GLCM	74
4.4.1. Założenia eksperymentu	74
4.4.2. Wyniki	77
4.4.3. Wnioski	79
Bibliografia	80

Wstęp

Monografia ta jest rezultatem wieloletnich badań w zakresie opracowania obrazów rastrowych — zarówno zdjęć lotniczych, jak i obrazów satelitarnych — przy użyciu szerokiego spektrum metod przetwarzania obrazów, a także statystycznych metod klasyfikacji. W szczególności do klasyfikacji szeroko wykorzystano sztuczne sieci neuronowe, których podstawowe zalety to:

- zdolność przetwarzania danych w sytuacjach, gdy brak jest poprawnie skonstruowanego modelu albo w ogóle dobrze zdefiniowanej wiedzy o problemie, lub jakiegokolwiek informacji heurystycznej,
- zdolność uczenia się algorytmów przetwarzania danych bez programowania, na podstawie serii przykładów,
- zdolność uogólniania,
- odporność na błędy i zakłócenia.

W ostatniej dekadzie systemy informacji przestrzennej w Polsce uległy znacznym zmianom, polegającym na tym, że zaczęto masowo wykorzystywać zdjęcia lotnicze i satelitarne. Jednym z czynników motywujących te zmiany było przyjęcie Polski do Unii Europejskiej i wdrażanie systemu IACS, które wymaga pokrycia całego kraju wysoko rozdzielczymi zdjęciami lotniczymi i częściowo satelitarnymi. Oprócz wykorzystania do celów kontroli bezpośrednich dopłat do rolnictwa znalazły one szereg innych zastosowań, w tym do budowy krajowej infrastruktury danych przestrzennych: bazy danych topograficznych TBD o szczegółowości odpowiadającej skali 1:10,000 oraz wektorowej mapy poziomu drugiego VMAP L2 o szczegółowości odpowiadającej skali 1:50,000.

Początkową przesłanką badań prowadzonych w ramach projektu* było opracowanie technologii wykorzystania sztucznych sieci neuronowych do zasilania i weryfikacji baz danych referencyjnych na podstawie zdjęć lotniczych i satelitarnych. W szeregu wstępnych eksperymentów została zastosowana metoda analizy obrazów za pomocą maski (otoczenia analizowanego piksela), stanowiącej dane wejściowe dla klasyfikatora — sieci neuronowej, sekcje 1.2 i 1.3. Ponieważ nie wszystkie wyniki otrzymywane za pomocą automatycznie uczonej sieci neuronowych były zadowalające, dla porównania zastosowano kilka podstawowych metod statystycznych, jak metody analizy skupień (sekcja 1.4) oraz klasyfikator bayesowski, tzw. naiwny (sekcja 1.5).

W prowadzonych badaniach bardzo szybko pojawiła się potrzeba zastosowania obiektywnych metod oceny jakości wyników otrzymywanych w kolejnych eksperymentach. Najbardziej przydatne — z punktu widzenia klasyfikacji obszarów — byłoby posiadanie obiektywnej i całościowej miary jakości klasyfikacji obrazów

*Niniejsze opracowanie powstało w ramach projektu badawczego KBN nr 5T12E02924.

przy zastosowaniu danej metody. Dlatego dokonano analizy znanych miar jakości tego typu i w sekcji 1.6 zawarto ich przegląd oraz omówienie. Okazało się, że żadna pojedyncza metoda oceny nie daje wymiernej oraz bezwzględnej miary jakości klasyfikacji, dlatego w dalszej części pracy przedstawiony jest cały zestaw tych ocen.

W dalszych pracach podjęto klasyczne zadanie klasyfikacji obrazów satelitarnych. W tym zakresie wykonano szereg eksperymentów z danymi z NOAA, Landsata i satelity SPOT. Rozszerzono zakres przetwarzanej informacji na dodatkowe cechy statystyczne obrazu, pozwalające wnioskować — oprócz odbicia spektralnego — o teksturze analizowanego obszaru. W tym celu zastosowano szereg metod ekstrakcji cech statystycznych obrazu, takich jak: momenty chromatyczne, pola losowe Markowa i Gibbsa, wariogramy oraz metodę macierzy GLCM. Podstawy teoretyczne tych metod, jak również dokładne opisy eksperymentów przeprowadzonych z ich zastosowaniem, opisano w rozdziałach 2 i 3. Wyniki osiągnięte w tych badaniach były zróżnicowane, na przykład segmentacja obrazu z wykorzystaniem momentów chromatycznych przyniosła częściowy sukces, natomiast próba zastosowania pól losowych Markowa nie powiodła się. Inne wyniki klasyfikacji z użyciem tekstur były bardzo zadowalające, z błędami względnymi na poziomie kilku procent.

Kolejny etap badań dotyczył przetwarzania zdjęć lotniczych (rozdział 4). Związane było to z intensyfikacją prac nad opracowaniem baz danych referencyjnych, dla których źródłem pozyskania danych są zdjęcia lotnicze. Celem prac nad zdjęciami lotniczymi nie była automatyzacja procesu pozyskania danych, a jedynie automatyzacja procesu rozpoznawania wybranych obiektów, która może wspomagać proces kontroli prac związanych z budową bazy danych przez firmy kartograficzne. Ponadto podjęto próby wyznaczenia obszarów zabudowanych dla TBD i VMAP poziomu 2. Ponieważ wyniki osiągnięte za pomocą sieci neuronowej tylko na podstawie odbicia spektralnego były niewystarczające, dla ich poprawy konieczne były dodatkowe przetwarzanie i zaawansowane analizy przestrzenne. W tym wypadku zastosowano opracowane wcześniej techniki wykorzystania tekstur, głównie metodę macierzy GLCM. To bardzo poprawiło osiągnięte wyniki klasyfikacji, aczkolwiek bezwzględna poprawność tej klasyfikacji zależała od typu rozpoznawanych obiektów, od doskonałych wyników dla klas takich jak wody i lasy do nieco gorszych rezultatów dla budynków i dróg. W większości przypadków uwzględnienie parametrów GLCM pozwalało na osiągnięcie wyników podobnych lub lepszych, niż było to możliwe za pomocą masek, przy jednoczesnym mniejszym nakładzie obliczeń.

W prowadzonych w całym projekcie pracach wykorzystano wyłącznie oprogramowanie niekomercyjne. Dotyczy to zarówno kompletnych, gotowych pakietów oprogramowania użytkowego, takich jak doskonały symulator sieci neuronowych SNNS oraz biblioteki procedur przetwarzania grafiki OpenCV i Image Magick, jak również podstawowego oprogramowania narzędziowego, takiego jak: kompilatory C i C++, edytory i debuggery, za pomocą których opracowano szereg własnych, autorskich programów wymienionych i opisanych w tym opracowaniu.

1. Klasyfikacja obszarów na obrazach satelitarnych i zdjęciach lotniczych

Tradycyjnie, najczęściej stosowaną metodą klasyfikacji obiektów na obrazach satelitarnych i zdjęciach lotniczych jest interpretacja wizualna (Kozubek, 2000). Metoda ta łatwo pokonuje takie trudności, jak wyróżnienie obiektów o dużym zróżnicowaniu odbicia spektralnego oraz odróżnienie obiektów o identycznej barwie, lecz innych teksturach powierzchni (Białach, 2000).

Probabilistyczne metody klasyfikacji obrazów satelitarnych najczęściej opierają się na założeniu, że rozkład prawdopodobieństwa odpowiedzi spektralnej obrazu jest znany, przy czym najczęściej przyjmuje się w tym celu rozkład Gaussa. W przypadku wielu obiektów rozpoznawanych na obrazach satelitarnych to założenie nie jest jednak słuszne, a rzeczywisty rozkład prawdopodobieństwa ich odbicia spektralnego nie jest znany. Obszary zabudowane dostarczają wielu przykładów takich obiektów, ponieważ stanowią połączenie asfaltu, betonu, roślinności i innych rodzajów pokrycia (Jakomulska, 1998). W procesie rozpoznawania takich obiektów przydatne okazuje się uwzględnienie informacji o strukturze i teksturze występujących tam obiektów.

W analizie zdjęć lotniczych i wielospektralnych obrazów satelitarnych często wyróżnia się dwa następujące po sobie etapy obróbki danych: klasyfikacja *per pixel*, gdzie dochodzi do klasyfikacji pojedynczych pikseli ze względu na rodzaj powierzchni, oraz klasyfikacja *per parcel*, gdzie dochodzi do klasyfikacji większych obszarów ze względu na charakter użytkowania. Jednak dwuetapowe przetwarzanie nie zawsze jest regułą. Zdarza się, że segmentacja zdjęcia na obszary różniące się pod względem sposobu użytkowania odbywa się jednoetapowo. Niepoślednią wtedy rolę odgrywają wiedza i umiejętności operatora dokonującego tej klasyfikacji.

Niniejsza monografia opisuje eksperymenty zainspirowane między innymi następującymi pytaniami:

- Czy można wykorzystać wyniki klasyfikacji jednoetapowej do automatycznej klasyfikacji obszarów pod względem użytkowania?
- Jakim odbiciem spektralnym charakteryzują się wyróżnione na jednym etapie klasy?
- Czy można zastąpić klasyfikację jednoetapową — dwuetapową?

1.1. Klasyfikacja *per pixel* i *per parcel*

Klasyfikacja *per pixel* polega na:

- wyróżnieniu pewnych kategorii (klas) pikseli odpowiadających różnym typom pokrycia powierzchni;
- zaklasyfikowaniu każdego z punktów analizowanego zdjęcia do właściwej mu klasy.

Decydującą rolę w procesie wyróżniania kategorii odgrywa indywidualny charakter odbicia spektralnego poszczególnych powierzchni. Wiadomym jest, że w określonym zakresie częstotliwości fal niektóre powierzchnie odpowiadają mocniej, w innych zaś słabiej. Właśnie te różnice pozwalają ocenić, z jakim konkretnym typem powierzchni ma się do czynienia.

Zakres widm, w jakich wykonuje się zdjęcia, zaprojektowano tak, aby wykrywanie klas pokrycia powierzchni jak najbardziej ułatwić. Istnieją jednak klasy, które mimo tego zabiegu trudno jest odseparować. Co więcej, na wyniki klasyfikacji wpłynąć mogą również warunki, w jakich wykonano zdjęcia (pora dnia, zachmurzenie itp.).

Zazwyczaj wyróżnianie kategorii odbywa się na podstawie zgromadzonej już wiedzy. Przeprowadzone doświadczenia pozwalają z góry przewidzieć, jakim odbiciem charakteryzować się będą konkretne typy powierzchni w poszczególnych pasmach spektralnych. Zatem klasyfikacja manualna ma charakter dalece arbitralny.

Innym sposobem wyróżnienia kategorii jest ich automatyczna selekcja. W tym przypadku kategorie są wyróżniane metodami analizy skupień. Określenie liczby wszystkich klas powierzone może być człowiekowi — (klasyfikacja z nauczycielem — *supervised classification*). Liczba klas może też być wyznaczona automatycznie (klasyfikacja bez nauczyciela — *unsupervised classification*).

Obydwa sposoby prowadzą do wyróżnienia kategorii definiujących klasy rozpoznawalnych na zdjęciu typów powierzchni. Klasyfikacja każdego z pikseli na zdjęciu do wyróżnionych kategorii prowadzi do otrzymania obrazu przedstawiającego sklasyfikowaną już powierzchnię analizowanego obszaru.

Sposób, w jaki dokonywana jest klasyfikacja pikseli, zależy od definicji kategorii. W najprostszym przypadku kategorie interpretowane są jako centra skupień (punkty w kostce n -wymiarowej), miarą zaś przynależności pikseli do tej samej klasy jest euklidesowa odległość od odpowiedniego centrum.

Klasyfikacja *per parcel* jest krokiem, który prowadzi od obrazu przedstawiającego typy powierzchni do obrazu przedstawiającego sposób zagospodarowania. Klasyfikacja *per parcel* polega na przeprowadzeniu lokalnej klasyfikacji (na arbitralnie zdefiniowanych obszarach) i zagregowaniu otrzymanych wyników w jeden wynik końcowy.

Przykładem ręcznej klasyfikacji *per parcel* jest uśrednianie do działki, polegające na nałożeniu maski z działkami katastralnymi i większościowe ujednocianie

całych działek (co jest poprawne, o ile jedna działka rolna jest równoważna jednej działce katastralnej).

Jedną z najprostszych realizacji automatycznej klasyfikacji *per parcel* jest filtracja medianowa, którą najczęściej dokonuje się dla maski o wymiarze 3×3 lub 5×5 pikseli. W bardziej skomplikowanych rozwiązaniach obliczane są lokalne statystyki dla określonego systemu sąsiedztwa, dokonuje się również filtracji częstotliwościowej itp.

Istnieją różne metody przeprowadzania bardziej zaawansowanej klasyfikacji *per parcel*. Biorąc pod uwagę sposób generowania cech podlegających klasyfikacji, wyróżnić można metody:

- operujące na obrazach rastrowych, generujące cechy jako wynik lokalnej analizy; do tej grupy zaliczyć można metody analizy tekstur, jak np.: metoda GLCM (Carr, 1996; Tuceryan and Anil, 1998; Berberoglu et al., 2000), metoda pól losowych Gibbsa i Markowa (Keuchel et al., 2003), metody fraktalne (Cheng, 1999), wariogram (Chica-Olmo and Abarca-Hernández, 2000); oraz metody przekształceń morfologicznych, jak np. algorytm *region growth* (Lira and Maletti, 2002);
- transformujące obraz rastrowy do reprezentacji obiektowej, dla której przeprowadzana jest analiza kontekstowa; do tej grupy należą np. systemy ekspertowe lub metody klasyfikacji kontekstowej (Barr and Barnsley, 1999);
- łączące podejście rastrowe i obiektowe, jak metody wiążące wektory z rastrami (Aplin and Atkinson, 2001).

Idea rozpoznawanie tekstur na obrazach rastrowych pojawiła się dość wcześniej (Haralick, 1979), owocując dojrzałymi rozwiązaniami w dziedzinie obróbki obrazów (Haralick and Shapiro, 1992). Doświadczenia pokazały, że wzbogacenie klasyfikacji *per pixel* o cechy teksturowe wydatnie polepsza wyniki klasyfikacji (Barr and Barnsley, 1999; Berberoglu et al., 2000; Carr, 1996; Clausi and Zhao, 2002; Haralick and Shapiro, 1992; Ryherd and Woodcock, 1996; Zhang, 1999).

Biorąc pod uwagę sposób budowy klasyfikatora, wyróżnić można metody nadzorowane i nienadzorowane, wykorzystujące: sztuczne sieci neuronowe (Miller et al., 1995; Liu et al., 2002; Ruiz-del Solar, 1998), klasyfikatory *k-means*, klasyfikatory maksymalnej wiarygodności, metody analizy dyskryminacyjnej, metody zbiorów rozmytych, (Abkar et al., 2000; Foody, 2000; Hubert-Moy et al., 2001; Liu et al., 2002; Ju et al., 2003; Keuchel et al., 2003; Pal and Mather, 2003; Stuckens et al., 2000; Venkatesh and Raja, 2003), oraz metoda wektorów nośnych (*Support Vector Machines*) (Bellman and Shortis, 2004).

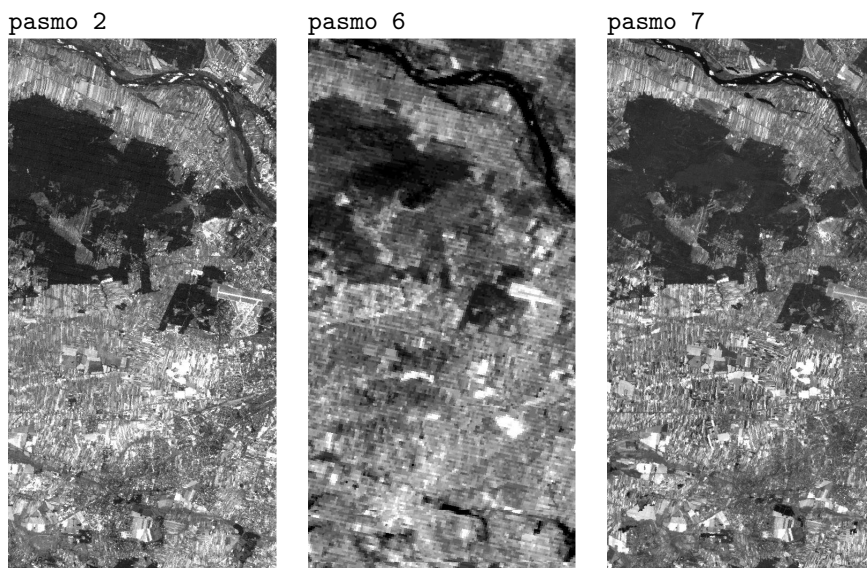
1.2. Obrazy Landsat TM

Obrazy satelitarne, wykorzystane w eksperymentach opisanych w tym rozdziale, pochodzą ze skanera TM (Thematic Mapper) satelity Landsat. Skaner TM

rejestruje energię odbitą od obiektów (oraz emitowaną przez nie samoistnie) w zakresie od pasma widzialnego do podczerwieni. Rozdzielczość przestrzenna tych obrazów wynosi 30×30 dla wszystkich pasm poza pasmem podczerwieni, które ma rozdzielczość przestrzenną 120×120 m. Dokładniej własności obrazów pochodzących z tego urządzenia przedstawia poniższa tabela:

	pasmo	długość fali (μm)	rozdzielczość (m)
niebieski	1	0.45–0.52	30
zielony	2	0.52–0.60	30
czerwony	3	0.63–0.69	30
bliska podczerwień	4	0.76–0.90	30
średnio bliska podczerwień	5	1.55–1.75	30
podczerwień termiczna	6	10.40–12.50	120/60
średnia podczerwień	7	2.08–2.35	30
panchromatyczny		0.5–0.9	15

Przykładowe obrazy Landsat TM, przedstawiające fragment miasta Warszawy i okolic, znajdują się na rysunku 1.1.

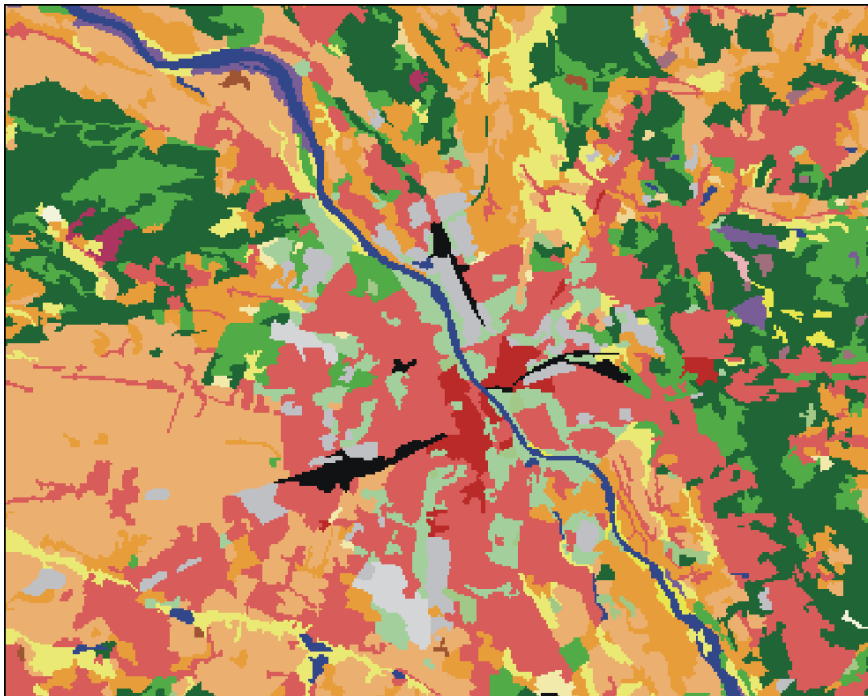


Rys. 1.1. Przykładowe obrazy Landsat TM okolic Warszawy: pasma 2, 6 i 7. Rozdzielczość pasm 2 i 7 wynosi 30×30 m, natomiast pasma 6 — 120×120 m [Landsat Inc., IGIK].

1.3. Projekt CORINE

Implementacja procesów klasyfikacji *per pixel* stawia określone wymagania wobec danych. Konieczne jest precyzyjne określenie klas obiektów, które mają być rozpoznane na zdjęciach, oraz zidentyfikowanie cech na obrazach spektralnych, które mogą pomóc w rozpoznaniu tych obiektów.

W opisanych tu eksperymentach wykorzystano opracowanie programu CORINE (*Coordination of Information on the Environment*) zainicjowanego przez Komisję Europejską w celu gromadzenia informacji o stanie środowiska w całej Europie. Baza danych tego projektu zawiera 44 klasy użytkowania terenu obejmujące 5 głównych kategorii (pierwszego poziomu): powierzchnie sztuczne, obszary rolnicze, lasy, naturalne i półnaturalne murawy, torfowiska i obszary wodne (Baranowski and Ciołkosz, 1997) (rys. 1.2).



Rys. 1.2. Mapa klas użytkowania terenu CORINE w skali 1:100,000
[Image&Corine Land Cover 2000, GIOŚ].

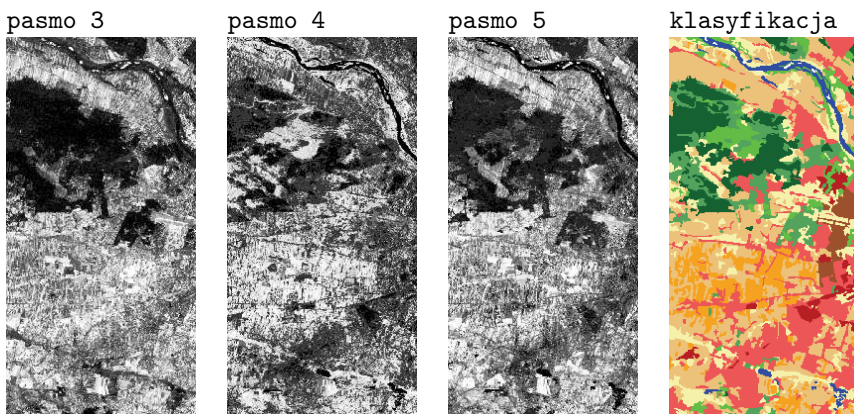
W trakcie prac prowadzonych z danymi CORINE okazało się, że ta struktura zawiera klasy obiektów zdefiniowane sztucznie i zalicza do różnych klas obiekty

o identycznym obrazie spektralnym, jak np. lasy liściaste i parki w miastach. Z drugiej strony, wymaga zaklasyfikowania jako jednej klasy grupy obiektów o zupełnie odmiennej charakterystyce spektralnej. Ponadto, opracowania CORINE, oryginalnie opracowane przez IGiK, zawierały szereg błędów i przeoczeń*. Dlatego dla potrzeb tej pracy dokonano ręcznej korekty fragmentu opracowania CORINE okolic Warszawy. Otrzymany poprawiony obraz, zawierający 11 klas o zróżnicowanej charakterystyce spektralnej, pozwolił uzyskać wiarygodne wyniki eksperymentów klasyfikacji automatycznej (rys. 1.2).

1.4. Metody analizy skupień

Obiektem badań w naszych eksperymentach były obrazy przedstawiające lewą stronę miasta Warszawy (rys. 1.3):

- trzy obrazy Landsat TM: pasmo 3 (99 kolorów, 33 różne), pasmo 4 (148 kolorów, 48 różnych) i pasmo 5 (181 kolorów, 55 różnych);
- jeden obraz z dwunastoma klasami odpowiadającymi różnym sposobom użytkowania (klasyfikację przeprowadzono manualnie, jak opisano powyżej).



Rys. 1.3. Obrazy satelitarne po wyrównaniu histogramów (pasmo 3, 4 i 5) oraz obraz z klasyfikacją obszarów ze względu na sposób użytkowania.

Wszystkie rastry mają rozmiar 500×100 pikseli i przedstawiają dokładnie ten sam obszar. Ponieważ kontrast obrazów satelitarnych był niewielki, przed wykonaniem właściwych eksperymentów poddano je obróbce wstępnej, polegającej na wyrównaniu histogramów.

*W trakcie przeprowadzania początkowych eksperymentów autorzy borykali się z dużymi błędami wyników obliczeń opartych na opracowaniach CORINE. Szczegółowe porównanie CORINE z innymi opracowaniami (w tym mapami topograficznymi) pozwoliło zidentyfikować CORINE jako źródło tych błędów.

1.4.1. Opracowanie danych

Przed rozpoczęciem właściwych eksperymentów wykonano wstępne opracowanie danych, polegające na utworzeniu i analizie histogramów odzwierciedlających rozkład poziomów szarości pikseli zakwalifikowanych do tej samej klasy w jedno-etapowym procesie klasyfikacji w kolejnych pasmach. Analizie poddano obrazy oryginalne. Wynik eksperymentu miał pokazać, jaka jest charakterystyka odbicia spektralnego pikseli należących do tej samej klasy.

1.4.1.1. Budowa histogramów

Do stworzenia histogramów użyto programu `histogramy`, którego argumentami były obrazy satelitarne (obrazy szare o 256 poziomach szarości) oraz obraz z naniesionymi klasami (obraz kolorowy). Program ten wypisuje:

- liczbę zaobserwowanych klas w pliku kolorowym (co dokładnie odpowiada liczbie kolorów na obrazie) wraz z opisem formatu danych;
- listę klas z przypisanymi kolorami; w naszym przykładzie wystąpiły następujące przypisania kolorów:

Klasa	1	2	3	4	5	6	7	8	9	10	11	12	
Kolor	R	241	252	250	86	0	255	86	194	0	255	160	0
	G	197	245	87	169	100	165	207	0	0	255	81	0
	B	121	171	87	92	0	0	48	0	255	255	44	0

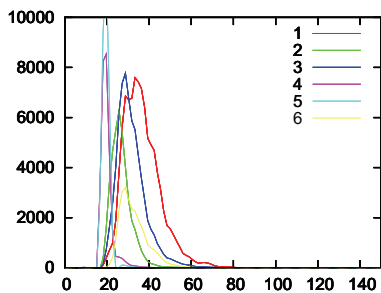
- informacje o zawartości obrazów szarych w postaci dwóch kolumn: pierwsza — to poziom szarości, druga — to liczba pikseli w klasie, które miały właśnie taki poziom szarości.

1.4.1.2. Wizualizacja histogramów

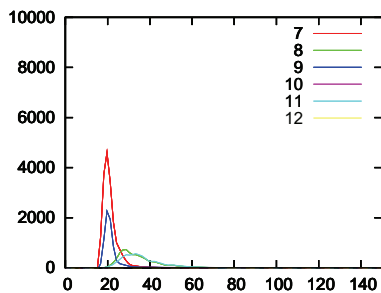
Do skonstruowania wykresów histogramów użyto programu `gnuplot`. Program ten ogranicza liczbę rozróżnialnych linii do 8, toteż liczba histogramów umieszczonych na jednym rysunku w naszym przypadku nie przekraczała tej wartości. Wyprodukowane przez `gnuplot` wykresy zawierały (rys. 1.4 i 1.5):

- wszystkie klasy, osobno dla każdego pasma (z podziałem na zbiory klas 1–6 i 7–12 wynikającym z ograniczenia na liczbę rozróżnialnych linii);
- każdą klasę osobno w 3 pasmach.

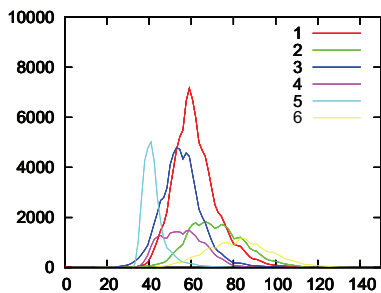
pasmo 3, klasy 1-6



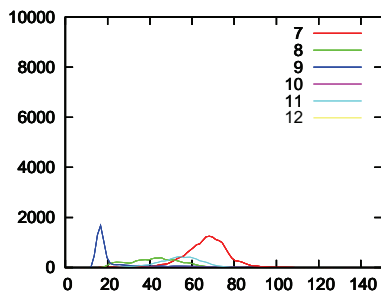
pasmo 3, klasy 7-12



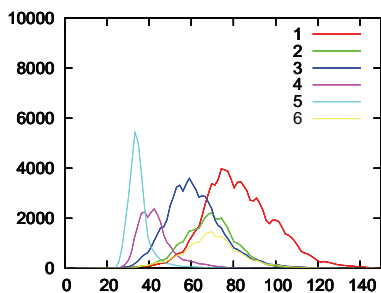
pasmo 4, klasy 1-6



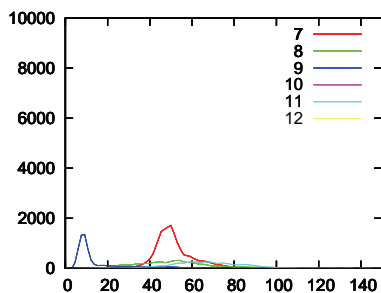
pasmo 4, klasy 7-12



pasmo 5, klasy 1-6

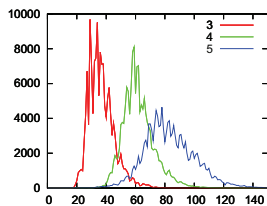


pasmo 5, klasy 7-12

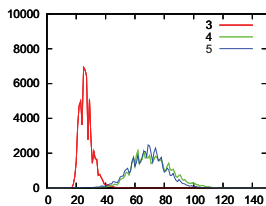


Rys. 1.4. Histogramy 12 klas CORINE w poszczególnych pasmach.

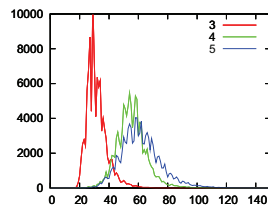
pasma 3,4,5 klasa 1



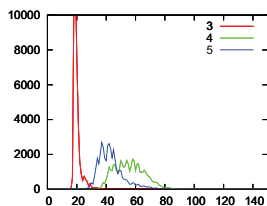
pasma 3,4,5 klasa 2



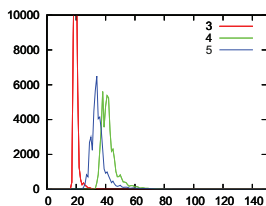
pasma 3,4,5 klasa 3



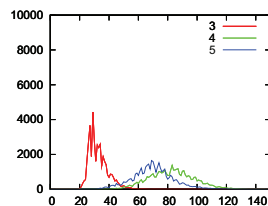
pasma 3,4,5 klasa 4



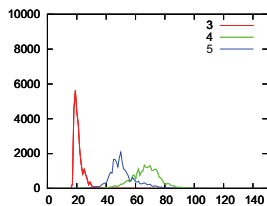
pasma 3,4,5 klasa 5



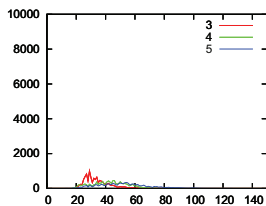
pasma 3,4,5 klasa 6



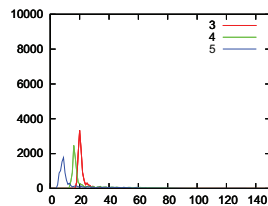
pasma 3,4,5 klasa 7



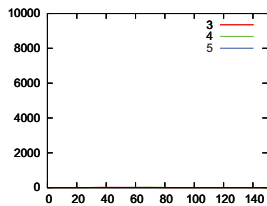
pasma 3,4,5 klasa 8



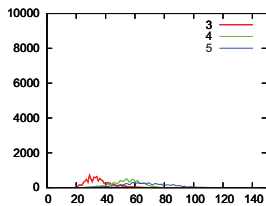
pasma 3,4,5 klasa 9



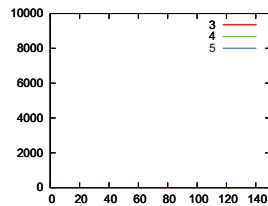
pasma 3,4,5 klasa 10



pasma 3,4,5 klasa 11



pasma 3,4,5 klasa 12



Rys. 1.5. Histogramy poszczególnych klas CORINE w pasmach 3,4,5.

1.4.1.3. Wstępne wnioski

Już pobieżna analiza skonstruowanych histogramów badanych klas wskazuje, że jakkolwiek we wszystkich analizowanych pasmach histogramy te mają różną postać, to jednak trudno na ich podstawie utworzyć prostą regułę klasyfikacji. Trudność wynika z faktu, iż w większości przypadków histogramy pokrywają się ze sobą.

1.4.2. Próby klasyfikacji *per pixel*

Eksperyment polegał na przeprowadzeniu automatycznej klasyfikacji *per pixel* na analizowanych obrazach i ocenie jakości tej klasyfikacji. Posłużono się narzędziami programowymi wchodzącymi w skład pakietu `fc-0.3.7`. Narzędzia stanowią implementację algorytmów analizy skupień opisanych w (Höppner et al., 1999). Szczegółowy opis pakietu `fc-0.3.7` znajduje się w jego dokumentacji (Höppner, 2000).

1.4.2.1. Przygotowanie danych

Narzędzia pakietu `fc-0.3.7` pracują na strumieniach danych w postaci tekstowej, dlatego samo przygotowanie danych zrealizowano w dwóch krokach. Na początku tworzono plik tekstowy za pomocą programu `obrazki2wektory`, a następnie programem `ddlimport` zmieniano jego format na postać akceptowalną przez inne narzędzia pakietu `fc-0.3.7`.

1.4.2.2. Automatyczna klasyfikacja

W pakiecie `fc-0.3.7` zaimplementowanych jest kilka algorytmów klasyfikacji działających na zasadzie analizy skupień (ang. *cluster analysis*), czyli poszukiwania zgrupowań wektorów danych* leżących blisko siebie i jednocześnie oddalonych od innych punktów, należących do innych skupień. Są to algorytmy:

- Algorytm *fuzzy c-means*, zaimplementowany w programie `fcm`, znajdujący sferyczne chmury danych.

Z założenia, skupienia powinny być podobnego rozmiaru. W każdym skupieniu wyróżnia się punkt centralny (punkt centralny — punkt w przestrzeni danych charakteryzujący dane skupienie). Przy ocenie przynależności danego wektora do danej klasy brana jest pod uwagę odległość euklidesowa tego wektora od punktu centralnego. Położenia pikseli centralnych wyliczane są jako średnie sumy ważone wektorów danych (gdzie wagami są stopnie przynależności wektorów do skupień).

*W tej prezentacji pojęcie *wektora danych*, czyli uporządkowanej n-ki jakichś wartości, będzie używane zamiennie z pojęciem punktu albo piksela obrazu, które również określone są n parametrami, takimi jak wartości spektralne.

- Algorytm *Gustafsona-Kessela*, zaimplementowany w programie `gk`, znajdujący elipsoidalne chmury danych.

Tak jak poprzednio, zakłada się w tym algorytmie podobny rozmiar każdego ze skupień. Przy ocenie przynależności danego wektora do danej klasy zmieniono miarę odległości od centrum. Tym razem jest to metryka określona pozytywnie zdefiniowaną macierzą (liczy się tzw. odległość Mahalanobisa).

- *Gath-Geva algorithm*, zaimplementowany w programie `gg`, zakładający, że wektory danych stanowią zbiorowisko c normalnych dystrybucji Gaussa. Dostarczane *a-priori* prawdopodobieństwo cech należących do normalnej dystrybucji może być interpretowane jako dodatkowy parametr mówiący o rozmiarze skupienia. Miarą odległości w tym przypadku jest kombinacja arbitralnej funkcji z odległością Mahalanobisa.
- *Quick clustering algorithm*, zaimplementowany w programie `qc`, służący do inicjalizacji danych. Ten algorytm jest podobny do algorytmu zaimplementowanego w `cini`. Dzieli on zbiór danych na równe podzbiory (pod względem liczności) i stara się znaleźć takie położenie punktów centralnych, aby odległość pomiędzy nimi była jak największa. Wynik działania algorytmu zależy od kolejności dostarczonych danych, czyli zwykle od przypadku.

Wszystkie wymienione wyżej algorytmy wyznaczają punkty centralne dla zadanej z góry liczby skupień. Co więcej, algorytmy te muszą startować z wprowadzonymi już punktami centralnymi. Dlatego też w celu znalezienia optymalnej liczby skupień dla analizowanych danych należy:

- uruchomić jeden z powyższych algorytmów przy różnych liczbach zainicjalizowanych punktów centralnych;
- spośród wygenerowanych wyników wybrać takie, które najlepiej spełniają postawione warunki (są najlepsze według operatora).

Inicjalizacji skupień można dokonać narzędziem `cini`. Narzędzie to pozwala wprowadzić jeden lub cały szereg zestawów skupień. Na przykład polecenie `cini -rN` generuje N centrów skupień, zaś polecenie `cini -rN:M` generuje szereg $N, N+1, \dots, M$ centrów skupień.

Jeśli inicjalizacja skupień odbyła się z opcją `-rN:M`, to nic nie stoi na przeszkodzie, aby dokonać analizy skupień za pomocą programu `fcm`. Program ten wyliczy położenia centrów skupień dla każdego zestawu osobno. Wybór optymalnego rozwiązania można powierzyć narzędziu `pick` (dokładniejszy opis znajduje się w dokumentacji pakietu `fc-0.3.7`).

1.4.3. Wizualizacja

Wynikiem klasyfikacji za pomocą narzędzia `fc-0.3.7` są pliki tekstowe w specjalnym formacie. Zawierają one wartości, które określają stopień przynależności

kolejnych punktów do kolejnych skupień. Stopień ten wyraża się liczbą z przedziału $\langle 0, 1 \rangle$.

Do celów wizualizacji w pakiecie `fc-0.3.7` udostępniono dodatkowe narzędzia `gsv` oraz `xsv`. Umożliwiają one wyświetlenie skupień w dwu- lub trójwymiarowym układzie współrzędnych. Aby zobaczyć efekt klasyfikacji w postaci obrazu (tj. zobaczyć na zdjęciu obszary należące do tej samej klasy), należy dokonać kolejnej zmiany formatu danych. Do zmiany formatu danych służy program `wektory2bdt`. Po zmianie formatu dane można już przeglądać przy użyciu programu do wizualizacji `testqt02`.

Program `testqt02` etykietuje każdy punkt kolorem odpowiadającym klasie, do jakiej został on przypisany. Ze względu na to, że jeden punkt może przynależać jednocześnie do więcej niż jednej klasy (co wyraża się niezerowym stopniem przynależności), w programie przyjęto strategię mieszania kolorów. Jeśli wartość określająca przynależność danego punktu do jakiejś klasy przekracza zadany próg (wartość progu jest parametrem, który można ustawić w programie), to kolor tej klasy sumuje się z kolorami innych klas, dla których próg ten również został przekroczony. Uśredniony wynik tej sumy definiuje ostateczny kolor danego punktu, z jakim pojawi się on na ekranie.

1.4.4. Wyniki

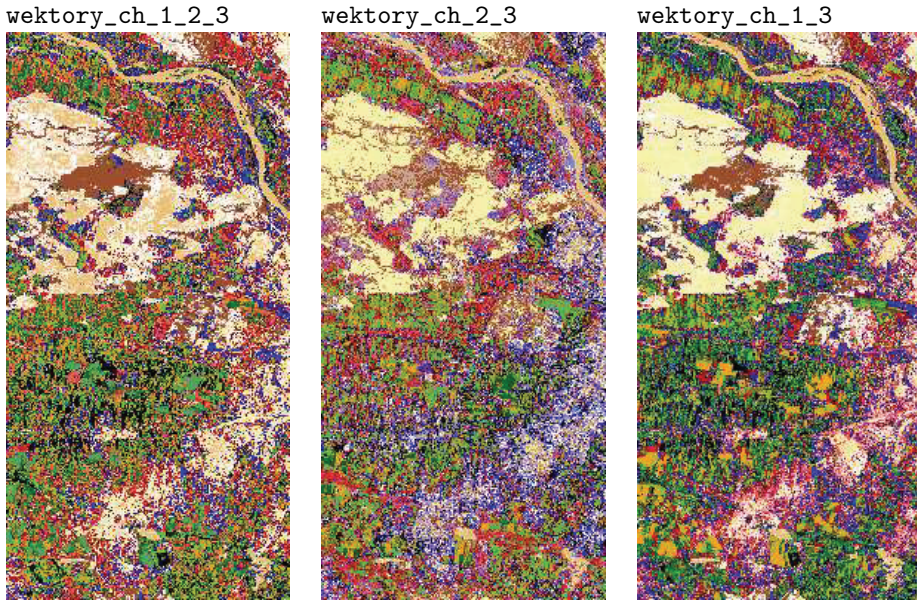
Do analizy za pomocą narzędzi `fc-0.3.7` przygotowane zostały następujące zestawy danych:

- `wektory_ch_1_2_3.ddl` — zawierający wektory danych dla pasm 3, 4, 5;
- `wektory_ch_2_3.ddl` — zawierający wektory danych dla pasm 4, 5;
- `wektory_ch_1_3.ddl` — zawierający wektory danych dla pasm 3, 5.

Wyniki analizy skupień uzyskano po wykonaniu poleceń:

- `cini -r12 < wektory_ch_1_2_3.ddl | fcm -o | save -q`
- `cini -r12 < wektory_ch_2_3.ddl | fcm -o | save -q`
- `cini -r8:22 < wektory_ch_1_3.ddl | fcm | pick | fcm -o | save -q`

po którym nastąpiła zamiana formatu danych i ich wizualizacja. Odpowiednie obrazy znaleźć można na rysunku 1.6. Jak łatwo zauważyć, wygenerowane rozwiązania dalekie są od oczekiwań. Obszary wód i lasów poddawały się klasyfikacji stosunkowo dobrze, natomiast pozostałe klasy pokrycia terenu — źle.



Rys. 1.6. Wyniki klasyfikacji programem fcm dla danych `wektory_ch_1_2_3`, `wektory_ch_2_3`, `wektory_ch_1_3` z wartością progu mieszania kolorów równą 0.5.

1.5. Naiwny klasyfikator bayesowski

1.5.1. Teoria Bayesa

Teoria Bayesa pozwala obliczyć prawdopodobieństwo *a posteriori*, $P(H|X)$. Prawdopodobieństwo to wyraża się równaniem:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1.1)$$

gdzie:

H — hipoteza (że X należy do klasy C);

X — przypadek (zestaw danych, fakty, obserwacje), które mogą wpłynąć na ocenę prawdopodobieństwa hipotezy;

$P(H)$ — prawdopodobieństwo *a priori*, że spełniona jest hipoteza H bez uwzględniania jakichkolwiek danych, np. $P(H)$ może być prawdopodobieństwem, że dowolny, losowo wybrany owoc (opisany wektorem danych) jest jabłkiem — nie zważając na jego kolor i kształt. Prawdopodobieństwo $P(H)$ nie zależy od X .

$P(H|X)$ — prawdopodobieństwo *a posteriori*, że hipoteza H jest prawdziwa dla zaobserwowanego przypadku X , np. prawdopodobieństwo, że owoc jest jabłkiem (H), jeśli znany jest warunek, że owoc ten jest okrągły i czerwony (X).

$P(X|H)$ — wiarygodność hipotezy H ze względu na X , np. mogłoby to być prawdopodobieństwo, że przypadkowo wybrany owoc X jest czerwony i okrągły, gdy wiadomo, że prawdziwa jest hipoteza, to jest, że X jest jabłkiem.

$P(X)$ jest prawdopodobieństwem wystąpienia danych X , np. mogłoby to być prawdopodobieństwo, że dany przypadek (zestaw danych) zarejestrowany wśród zbioru owoców jest czerwony i okrągły, ang. *evidence*.

Jeśli X jest ciągłą zmienną losową, której dystrybucja zależy od H_i i jest wyrażona przez warunkową funkcję gęstości rozkładu prawdopodobieństwa $p(X|H_i)$, regułę Bayesa można również przedstawić w postaci:

$$P(H_i|X) = \frac{p(X|H_i)P(H_i)}{p(X)} \quad (1.2)$$

gdzie $p(\cdot)$ to odpowiednie funkcje gęstości prawdopodobieństwa.

Dla dwóch hipotez dotyczących tego samego stanu rzeczy:

$$p(x) = \sum_{i=1}^2 p(X, H_i)P(H_i) \quad (1.3)$$

Jeśli dana jest obserwacja X oraz $P(H_1|X)$ jest większe od $P(H_2|X)$, to można byłoby wnioskować, że prawdziwą hipotezą jest H_1 . Prawdopodobieństwo popełnienia błędu wyraża się wtedy przez:

$$P(error|x) = \begin{cases} P(H_1|X) & \text{jeśli zdecydowano, że prawdziwe jest } H_2 \\ P(H_2|X) & \text{jeśli zdecydowano, że prawdziwe jest } H_1. \end{cases} \quad (1.4)$$

Można dowieść, że reguła wyboru hipotezy:

$$\text{Wybierz } H_1, \text{ jeśli } P(H_1|X) > P(H_2|X); \text{ w przeciwnym razie wybierz } H_2 \quad (1.5)$$

minimalizuje średnie prawdopodobieństwo popełnienia błędu $P(error)$:

$$P(error) = \int_{-\infty}^{\infty} P(error, X)dX = \int_{-\infty}^{\infty} P(error|X)p(X)dX \quad (1.6)$$

Mianownik w równaniu 1.2, jak można zauważyć, jest współczynnikiem skalującym, zapewniającym $P(H_1|X) + P(H_2|X) = 1$. Można więc stosować alternatywną postać reguły decyzyjnej, w której ten współczynnik już nie występuje:

$$\text{Wybierz } H_1, \text{ jeśli } p(X|H_1) > p(X|H_2); \text{ w przeciwnym razie wybierz } H_2 \quad (1.7)$$

1.5.2. Wielowymiarowa, ciągła przestrzeń cech

Następnie wprowadzimy oszacowanie ryzyka związanego z podejmowaniem decyzji w przypadku losowym. Niech $\omega_1, \dots, \omega_k$ odpowiadają k stanom natury (hipotezom) oraz $\alpha_1, \dots, \alpha_a$ będzie zbiorem możliwych akcji. W takim przypadku funkcja straty:

$$\lambda(\alpha_i|\omega_j) \quad (1.8)$$

opisuje stratę, jaką wywoła podjęcie akcji α_i , gdy stan natury jest ω_j .

Reguła Bayesa dla przypadku wielowymiarowej, ciągłej przestrzeni cech i dla wielu stanów natury wyraża się wzorem:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x})} \quad (1.9)$$

gdzie

$$p(\mathbf{x}) = \sum_{j=1}^k p(\mathbf{x}|\omega_j)P(\omega_j). \quad (1.10)$$

Oczekiwana strata (ryzyko warunkowe), związana z podjęciem akcji α_i , jest dana przez:

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^k \lambda(\alpha_i|\omega_j)P(\omega_j|\mathbf{x}). \quad (1.11)$$

Oczekiwana strata związana z regułą decyzyjną wyraża się równaniem:

$$R = \int R(\alpha(\mathbf{x})|\mathbf{x})p(\mathbf{x})d\mathbf{x} \quad (1.12)$$

gdzie $\alpha(\mathbf{x})$ jest funkcją decyzyjną, która dla każdego \mathbf{x} zakłada jedną z a wartości $\alpha_1, \dots, \alpha_a$.

Reguła decyzyjna w takim przypadku polega na minimalizacji ryzyka całkowitego, tj. na obliczeniu ryzyka warunkowego:

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^k \lambda(\alpha_i|\omega_j)P(\omega_j|\mathbf{x}). \quad (1.13)$$

dla $i = 1, \dots, a$ i wybraniu akcji α_i , dla której $R(\alpha_i|\mathbf{x})$ osiąga minimum. Minimum takie nosi nazwę ryzyka Bayesa. Oznacza się je przez R^* i jest to najlepszy wynik, jaki można osiągnąć.

1.5.3. Naiwny klasyfikator Bayesa

Naiwny klasyfikator Bayesa jest metodą klasyfikacji powstałą na bazie teorii Bayesa. Zalicza się ją do grupy metod uczenia maszynowego. Określenie „naiwny” odnosi się do faktu, iż w zastosowanym modelu prawdopodobieństwa przyjęto pełną niezależność zmiennych losowych, a to w rzeczywistości często mijają się z prawdą.

W myśl tej teorii prawdopodobieństwo, że zaobserwowany przypadek (o cechach X_1, \dots, X_n) należy do klasy C , wyraża się wzorem:

$$P(C|X_1, \dots, X_n) = \frac{P(C)P(X_1, \dots, X_n|C)}{P(X_1, \dots, X_n)} \quad (1.14)$$

Mianownik w powyższym równaniu nie zależy od C , a ponadto dane są wartości cech X_i , toteż przyjmuje się, że mianownik ten jest wartością stałą. Licznik zaś przedstawić można jako:

$$\begin{aligned} P(C, X_1, \dots, X_n) &= P(C)P(X_1, \dots, X_n|C) \\ &= P(C)P(X_1|C)P(X_2, \dots, X_n|C, X_1) \\ &= P(C)P(X_1|C)P(X_2|C, X_1)P(X_3, \dots, X_n|C, X_1, X_2) \\ &= \dots \end{aligned}$$

Z „naiwnego” założenia, że nie istnieje warunkowa zależność pomiędzy X_i oraz X_j ($i \neq j$), mamy:

$$P(X_i|C, X_j) = P(X_i|C) \quad (1.15)$$

a więc

$$P(C, X_1, \dots, X_n) = P(C)P(X_1|C)P(X_2|C)P(X_3|C) \dots \quad (1.16)$$

Stąd

$$P(C|X_1, \dots, X_n) = Z * P(C) \prod_{i=1}^n P(X_i|C) \quad (1.17)$$

gdzie Z — współczynnik skali zależny tylko od X_1, \dots, X_n (stała, jeżeli wartości cech są znane).

Naiwny klasyfikator Bayesa łączy przedstawiony model prawdopodobieństwa z regułą decyzyjną. Zazwyczaj regułą tą jest wybór najbardziej prawdopodobnej hipotezy. Regułą tę nazywa się regułą MAP (*maximum a posteriori*). Odpowiadający jej klasyfikator jest funkcją *classify* zdefiniowaną jak niżej:

$$\text{classify}(x_1, \dots, x_n) = \operatorname{argmax}_{c_k} P(C = c_k) \prod_{i=1}^n P(X_i = x_i|C = c_k) \quad (1.18)$$

gdzie $c_k, k = 1, \dots, K$ to jedna z klas. W uczeniu z nauczycielem parametry modelu są estymowane. Z założenia o niezależności cech wystarczy estymować wartości prawdopodobieństw (*class prior conditional feature model*), używając metody *maximum likelihood*, *Bayesian inference* lub innych.

1.5.4. Klasyfikacja bayesowska *per pixel*

Problem polega tu na stwierdzeniu, do jakiej klasy należy punkt obrazu, gdy znane są odpowiadające mu poziomy szarości w kilku pasmach. Aby przeprowadzić klasyfikację naiwnym klasyfikatorem Bayesa, należy najpierw ten klasyfikator zbudować. Klasyfikator buduje się na bazie założeń:

- dany jest obraz wzorcowy z naniesionymi K klasami oraz obrazy odpowiadające wzorcowi w n pasmach z 256 poziomami szarości (obrazy uczące);

- szukany jest sposób klasyfikacji punktów innego obszaru, gdy znane są odpowiadające mu obrazy spektralne (obrazy testowe).

Po zbudowaniu klasyfikatora Bayesa na podstawie wzorców można go zastosować do klasyfikacji punktów na obrazach testowych.

Budowę klasyfikatora (na bazie obrazów uczących) rozpoczyna obliczenie prawdopodobieństwa $P(C = c_k)$. Aproxymowane jest ono licznością klasy k (liczbą punktów zaklasyfikowanych do tej klasy na wzorcu, wyrażoną przez N_k), podzieloną przez liczbę wszystkich punktów obrazu.

$$P(C = c_k) = \frac{N_k}{\sum_{k=1}^n N_k} \quad (1.19)$$

Następnie rozkładami normalnymi aproxymowane są prawdopodobieństwa warunkowe (prawdopodobieństwo, że poziom szarości punktu w kanale i wynosi x_i , gdy wiadomo, że punkt należy do klasy c_k):

$$P(X_i = x_i | C = c_k) \cong f(x_{k,i}) \quad (1.20)$$

Rozkłady normalne zdefiniowane są funkcją:

$$f(x_{k,i}) = \frac{1}{\sigma_{k,i} \sqrt{2\pi}} e^{-\frac{(x_{k,i} - \mu_{k,i})^2}{2\sigma_{k,i}^2}} \quad (1.21)$$

gdzie $x_{k,i}$ — poziom szarości punktu, $k = 1, \dots, K$ — indeks odpowiadający klasie, $i = 1, \dots, n$ — indeks odpowiadający pasmu.

Odchylenie standardowe przypadku (będące aproxymacją $\sigma_{k,i}$) wynosi:

$$S_{N_k-1,i} = \sqrt{\frac{1}{N_k - 1} \sum_{j=1}^{N_k} (x_{k,i,j} - \bar{x}_{k,i})^2} \quad (1.22)$$

gdzie: $x_{k,i,j}$ — poziom szarości j -tego punktu z pasma i należącego (zgodnie z wzorcem) do klasy k .

Wartość średnia (będąca aproxymacją $\mu_{k,i}$) dana jest przez:

$$\bar{x}_{k,i} = \frac{1}{N_k} \sum_{j=1}^{N_k} x_{k,i,j} \quad (1.23)$$

Powyższe wzory definiują ostatecznie postać budowanego naiwnego klasyfikatora Bayesa.

1.5.5. Eksperymenty i wyniki

Do klasyfikacji *per pixel* obrazów satelitarnych naiwnym klasyfikatorem Bayesa użyły dwa opracowane programy: `histogramyBayes` oraz `testBayes`. Pierwszy

program z tej pary ma za zadanie zbudować klasyfikator na podstawie obrazów uczących oraz przeprowadzić klasyfikację zbudowanym klasyfikatorem. Aby tego dokonać, program `histogramyBayes` wyświetla wartości parametrów, na bazie których zbudowany został klasyfikator:

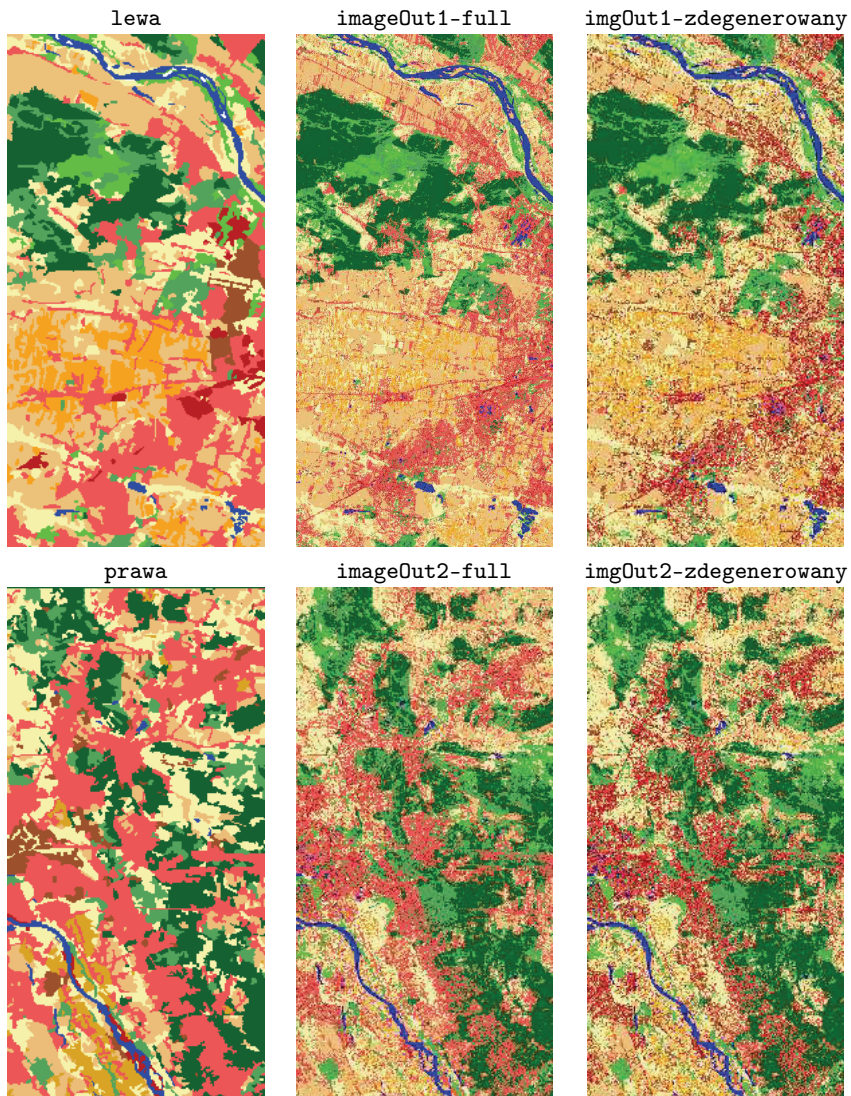
- liczbę zaobserwowanych klas na obrazie wzorcowym, co dokładnie odpowiada liczbie kolorów na obrazie,
- listę klas wraz z przypisanymi im kolorami,
- odchylenia standardowe i wartości średnie dla kolejnych klas w kolejnych pasmach, oraz licznosci poszczególnych klas.

Program następnie tworzy kolorowy obraz przedstawiający wyniki klasyfikacji. Dla zestawu obrazów `lewa`, `pasmo 3`, `pasmo 4`, `pasmo 5` (rys. 1.3) program `histogramyBayes` wyświetla następujące wartości:

kla -sa	kolor			wartość średnia			odchyl. standardowe			liczność
	r	g	b	pas3	pas4	pas5	pas3	pas4	pas5	
1	241	197	121	36.67	62.00	83.06	8.64	17.01	20.26	145411
2	252	245	171	26.82	71.64	68.35	3.10	14.37	9.97	60984
3	250	87	87	31.69	55.47	61.84	6.22	13.36	14.06	103769
4	86	169	92	20.00	55.87	43.44	1.53	8.42	5.77	41102
5	0	100	0	19.68	42.33	35.31	1.05	5.25	4.25	48334
6	255	165	0	32.76	82.68	69.91	3.52	12.61	8.81	40911
7	86	207	48	21.11	66.89	51.49	1.53	7.01	5.03	26597
8	194	0	0	34.11	42.45	52.53	2.45	5.11	5.88	11269
9	0	0	255	21.70	25.46	18.11	1.28	7.27	5.96	10905
10	255	255	255	49.62	63.49	93.10	1.08	1.77	3.01	548
11	160	81	44	35.80	54.16	67.49	2.37	4.06	5.17	10158
12	0	0	0	24.00	52.00	44.00	0.00	0.03	0.01	10

Program `testBayes` można wykorzystać do klasyfikacji *per pixel* obszarów testowych. Aby tego dokonać, należy program zasilić danymi koniecznymi do zbudowania klasyfikatora (wyprodukowanymi przez program `histogramyBayes`) oraz obrazami, które mają być klasyfikowane.

Wyniki zastosowania klasyfikatora bayesowskiego zbudowanego na podstawie analizy zdjęć lewej połowy Warszawy przedstawia rysunek 1.7. Klasyfikacji poddawane były obrazy przedstawiające prawą połowę Warszawy.



Rys. 1.7. Wyniki klasyfikacji „naiwnym” klasyfikatorem Bayesa. *lewa* to wzorek, który został użyty (razem z obrazami w 3 pasmach spektralnych) do stworzenia klasyfikatora.

Dla danych uczących otrzymano wyniki widoczne na *imageOut1-full* oraz *imageOut1-zdegenerowany*. *prawa* to wzorek dostarczony dla porównania (nie był użyty do zbudowania klasyfikatora). Obrazy *imageOut2-full* oraz *imageOut2-zdegenerowany* przedstawiają wyniki klasyfikacji dla obrazu testowego (w 3 pasmach). Przyrostek *full* oznacza, że klasyfikator został zaimplementowany zgodnie z regułą 1.18 a *zdegenerowany* oznacza, że w klasyfikatorze przyjęto $P(C = c_k) = 1$ dla wszystkich klas.

1.6. Ocena wyników klasyfikacji

1.6.1. Pierwiastek błędu średniokwadratowego ($RMSE$)

Pierwiastek błędu średniokwadratowego ($RMSE$) (Ju et al., 2003):

$$RMSE_k = \sqrt{\sum_{l=1}^L (\hat{N}_{kl} - N_{kl})^2 \times \frac{m_l}{M}} \quad (1.24)$$

gdzie \hat{N}_{kl} i N_{kl} to, odpowiednio, licznosc punktów przypisanych do klasy k wg klasyfikatora oraz licznosc punktów przypisanych do klasy k wg wzorca, przy czym klasyfikacja dotyczy obszaru l (obszar l może być zwartą powierzchnią na obrazie lub też zbiorem punktów wybranych losowo); m_l jest rozmiarem obszaru l wyrażającym się liczbą zawartych w obszarze punktów; M jest całkowitym rozmiarem danych, $M = \sum_{l=1}^L m_l$ (gdzie L — liczba wyróżnionych obszarów).

Uwaga: miarę powyższą można użyć do oceny wyników klasyfikacji dla jednego obrazu (wtedy wybiera się w nim obszary do wyliczenia $RMSE$) lub też do oceny wyników klasyfikacji dla wielu obrazów (wtedy każdy z nich jest osobnym obszarem).

1.6.2. Progowany błędem procent poprawnej klasyfikacji (TCE)

Progowany błędem procent poprawnej klasyfikacji (TCE) (Ju et al., 2003):

$$TCE_\epsilon = \frac{1}{K} \sum_{k=1}^K \sum_{l=1}^L n_{kl} \times \frac{m_l}{M} \quad (1.25)$$

gdzie K to liczba klas, zaś dla danego błędu ϵ :

$$n_{kl} = \begin{cases} 1 & \text{jeśli } |\hat{N}_{kl} - N_{kl}| \leq \epsilon \\ 0 & \text{w przypadku przeciwnym} \end{cases} \quad (1.26)$$

1.6.3. Współczynniki obliczane na podstawie macierzy niezgodności

Niech A będzie macierzą zawierającą wyniki klasyfikacji:

$$A = [a_{ij}] \quad (1.27)$$

gdzie a_{ij} jest liczbą punktów należących do klasy j , które zostały zaklasyfikowane jako punkty należące do klasy i . W przypadku oceny wyników klasyfikacji dla jednej tylko klasy (punkty nie należące do klasy „tworzą” drugą klasę) tabela ta może mieć postać:

		wg wzorca należy do klasy	
		tak	nie
wg klasyfikatora	tak	a_{11}	a_{12}
należy do klasy	nie	a_{21}	a_{22}

W przypadku wielu klas tabela ta może mieć postać:

		wg wzorca należy do klasy			
		1	2	...	K
wg klasyfikatora	1	a_{11}	a_{12}	...	a_{1K}
	2	a_{21}	a_{22}	...	a_{2K}

	K	a_{K1}	a_{K2}	...	a_{KK}

Na podstawie macierzy A obliczać można następujące szereg kolejnych współczynników (Hubert-Moy et al., 2001; Cohen, 1960) opisanych poniżej.

Dokładność użytkownika klasy i

$$u_i = a_{ii}/a_{ri}, \quad (1.28)$$

gdzie $a_{ri} = \sum_j a_{ij}$ (suma elementów w wierszu i);

Dokładność producenta klasy i

$$p_i = a_{ii}/a_{ci}, \quad (1.29)$$

gdzie $a_{ci} = \sum_j a_{ji}$ (suma elementów w kolumnie i);

Całkowita dokładność klasyfikacji d

$$d = \sum_i a_{ii}/a_t, \quad (1.30)$$

gdzie $a_t = \sum_i a_{ci} = \sum_i a_{ri}$ (liczba wszystkich punktów);

Prosty współczynnik Kappa

$$\hat{\kappa} = \frac{P_o - P_e}{1 - P_e}, \quad \text{gdzie } P_o = \frac{\sum_i a_{ii}}{a_t} \quad \text{oraz } P_e = \frac{\sum_i a_{ri} a_{ci}}{a_t^2} \quad (1.31)$$

Ważony współczynnik Kappa

$$\hat{\kappa}_w = \frac{P_{o(w)} - P_{e(w)}}{1 - P_{e(w)}}, \quad \text{gdzie } P_{o(w)} = \frac{1}{a_t} \sum_i \sum_j w_{ij} a_{ij} \quad (1.32)$$

$$\text{oraz } P_{e(w)} = \frac{1}{a_t^2} \sum_i \sum_j w_{ij} a_{ri} a_{cj}$$

Wagi w_{ij} spełniać powinny warunek: $0 \leq w_{ij} < 1$. Zazwyczaj przyjmuje się, że $w_{ii} = 1$ (czyli najważniejsza jest zgodność dotycząca tych samych klas), natomiast dla $i \neq j$ przyjmuje się $w_{ij} = w_{ji} = 1 - \frac{|i-j|}{K-1}$. Taka postać współczynników wagowych jest słuszna jedynie wtedy, gdy klasy uporządkowane są w tabeli pod względem ważności. Ważność danej klasy można ocenić na podstawie jej liczności.

Ponieważ wartość ważonego współczynnika Kappa zależy od doboru wag, nie ma prostej interpretacji otrzymywanych wartości. Zgodnie z jedną z propozycji, (Fleiss, 1981):

$\kappa > 0.75$	mocna zgodność
$0.4 < \kappa \leq 0.75$	dobra zgodność
$\kappa \leq 0.4$	słaba zgodność

Zgodnie z inną propozycją (Landis and Koch, 1977):

$0.81 \leq \kappa \leq 1$	prawie doskonała
$0.61 \leq \kappa \leq 0.80$	istotna
$0.41 \leq \kappa \leq 0.60$	umiarkowana
$0.21 \leq \kappa \leq 0.40$	rozsądna
$0.00 \leq \kappa \leq 0.20$	delikatna
$\kappa < 0.0$	słaba

(Nie)dokładność DA

$$DA = \frac{a_{12} + a_{21}}{a_t} \quad (1.33)$$

Miara ta wyraża stosunek sumy źle zaklasyfikowanych punktów dla każdej z klas do całkowitej ich liczby. $0 \leq uDA \leq 1$. Można zauważyć, że DA to „przeciwnieństwo” d dla macierzy o rozmiarze 2×2 . Dla większej macierzy można obliczyć uogólnioną (nie)dokładność uDA :

$$uDA = \frac{\sum_i (a_{ri} - a_{ii})}{a_t} = 1 - d \quad (1.34)$$

(Nie)dokładność DW

$$err1 = \begin{cases} \frac{a_{12}}{a_{11}+a_{12}} & \text{gdy } a_{11} + a_{12} > 0 \\ 0 & \text{w przypadku przeciwnym} \end{cases}$$

$$err2 = \begin{cases} \frac{a_{21}}{a_{21}+a_{22}} & \text{gdy } a_{21} + a_{22} > 0 \\ 0 & \text{w przypadku przeciwnym} \end{cases} \quad (1.35)$$

$$DW = \frac{err1+err2}{2}$$

Można zauważyć, że DW to „przeciwnieństwo” u_i dla macierzy o rozmiarze 2×2 . Dla większej macierzy można obliczyć uogólnioną (nie)dokładność uDW :

$$err_{ij} = \begin{cases} \frac{a_{ij}}{a_{ri}} & \text{gdy } a_{ri} > 0 \text{ oraz } i \neq j \\ 0 & \text{w przypadku przeciwnym} \end{cases}$$

$$err_i = \sum_j err_{ij} = \begin{cases} 1 - \frac{a_{ii}}{a_{ri}} = 1 - u_i & \text{gdy } a_{ri} > 0 \\ 0 & \text{w przypadku przeciwnym} \end{cases} \quad (1.36)$$

$$uDW = \frac{1}{K} \sum_i err_i$$

2. Segmentacja obszarów metodą momentów chromatycznych

Klasyfikacja obszarów metodą momentów chromatycznych pierwotnie służyła do segmentacji kolorowych zdjęć dostarczonych w formacie RGB. W metodzie tej dla zadanego obszaru obliczane były momenty chromatyczne, które następnie interpretowane były jako wektor cech analizowanych w procesie klasyfikacji. Metoda ta została przystosowana do segmentacji obrazów satelitarnych.

Obrazy satelitarne rozpatrywane w opisywanym eksperymencie zawierały obszary o różnym typie pokrycia powierzchni. Choćby obrazy niosły informacje w czterech różnych kanałach spektralnych, w opisywanym eksperymencie skorzystano z danych pochodzących tylko z trzech pierwszych kanałów. Kanały te zinterpretowano jako kolejne składowe RGB obrazu kolorowego.

2.1. Podstawy matematyczne

2.1.1. Konwersja RGB do CIE XYZ

Istnieje kilka definicji przestrzeni kolorów CIE XYZ. W opisywanych tu eksperymentach przyjęto, że konwersji przestrzeni kolorów RGB do CIE XYZ dokonuje się według wzoru:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.111 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

2.1.2. Chromatyczność

Chromatyczność, pomijając fizyczną interpretację, definiują następujące wzory:

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad z = \frac{Z}{X + Y + Z} \quad (2.2)$$

Mając x, y , można obliczyć z jako uzupełnienie do 1: $z = 1 - x - y$. Tak więc z jest wartością nadmiarową. Co więcej, wszystkie współrzędne mieszczą się w przedziale od 0 do 1: $x, y, z \in [0, 1]$.

2.1.3. Dystrybucja chromatyczności oraz jej ślad

Niech $I(i, j)$ oznacza punkt (piksel) obrazu I o współrzędnych i, j , przy czym $0 \leq i \leq L_x - 1$ oraz $0 \leq j \leq L_y - 1$, gdzie L_x, L_y — szerokość i wysokość obrazu. Dystrybucję chromatyczności definiuje poniższy wzór:

$$D(x, y) = K_{xy}, \quad (2.3)$$

gdzie K_{xy} to liczba pikseli, którym odpowiada chromatyczność (x,y) . Inaczej zapisując:

$$D(x, y) = \text{card}\{(i, j) \mid I(i, j) \rightarrow (x, y)\} \quad (2.4)$$

Ślad dystrybucji jest rzutem $D(x, y)$ na płaszczyznę x, y . Wyraża się on przez:

$$T(x, y) = \begin{cases} 1, & \text{jeśli istnieje piksel, któremu odpowiada chromatyczność}(x, y) \\ 0, & \text{w przeciwnym wypadku.} \end{cases} \quad (2.5)$$

Inaczej zapisując:

$$T(x, y) = \begin{cases} 1, & \exists(i, j) : I(i, j) \rightarrow (x, y) \\ 0, & \neg\exists(i, j) : I(i, j) \rightarrow (x, y) \end{cases} \quad (2.6)$$

Obliczanie momentów chromatycznych (patrz podrozdział 2.1.4) wymaga zmiany skali i dyskretyzacji. Niech $X_s, Y_s \in \aleph$ będą współczynnikami skalującymi. Niech $x_s = \text{Int}(X_s x)$, $y_s = \text{Int}(Y_s y)$, gdzie $x_s, y_s \in \aleph$ oraz Int jest funkcją zwracającą część całkowitą swojego argumentu. Przeskalowana (dyskretna) dystrybucja i jej ślad mają teraz postać:

$$D_s(x_s, y_s) = \text{card}\{(i, j) \mid I(i, j) \xrightarrow{x_s, y_s} (x_s, y_s)\} \quad (2.7)$$

$$T_s(x_s, y_s) = \begin{cases} 1, & \exists(i, j) : I(i, j) \xrightarrow{x_s, y_s} (x_s, y_s) \\ 0, & \neg\exists(i, j) : I(i, j) \xrightarrow{x_s, y_s} (x_s, y_s) \end{cases} \quad (2.8)$$

2.1.4. Momenty chromatyczne

Przy danej dystrybucji chromatyczności i danym śladzie dystrybucji można obliczyć momenty chromatyczne M_D (momenty dystrybucji) i M_T (momenty śladu dystrybucji). Momenty te definiują wzory:

$$M_D(k, l) = \sum_{x_s} \sum_{y_s} x_s^k y_s^l T_s(x_s, y_s) \quad (2.9)$$

$$M_T(k, l) = \sum_{x_s} \sum_{y_s} x_s^k y_s^l D_s(x_s, y_s) \quad (2.10)$$

2.2. Metoda segmentacji

Segmentacja obrazu, tj. klasyfikowanie punktów obrazów do właściwych klas, odbywała się na podstawie wartości momentów chromatycznych obliczonych dla lokalnych otoczeń tych punktów. Do tego celu opracowany został program `mrf_texture`. W programie tym lokalne otoczenie danego punktu definiowało położenie kwadratowej maski o zadanym rozmiarze, zaczepionej centralnie w tym

punkcie. Dla obszaru pokrytego maską wyliczane były momenty chromatyczne. Ich wartości były parametrami, na podstawie których odbywała się klasyfikacja.

Klasyfikatorem wykorzystanym do tych eksperymentów mógłby być dowolny klasyfikator, który na podstawie wektora danych (w tym wypadku wartości momentów chromatycznych) dokonuje poprawnej klasyfikacji obszaru. W badaniach, których wyniki są tu przedstawione, posłużono się klasyfikatorem najbliższego sąsiedztwa. Aby zbudować taki klasyfikator, opracowany program wyposażono w możliwość wczytywania obrazów referencyjnych. Na ich podstawie obliczano momenty chromatyczne, które można było zapisać do pliku. Wartości tych momentów służyły do zbudowania klasyfikatora, który mógł być użyty podczas kolejnego uruchomienia programu — w celu przeprowadzenia klasyfikacji. W tym przypadku obrazy referencyjne nie odgrywały już żadnej roli. Cała informacja, jaką one niosły, została zapisana w wartościach momentów chromatycznych obliczonych podczas poprzedniego uruchomienia programu (co przełożyło się na parametry zbudowanego klasyfikatora).

Ostatecznym wynikiem klasyfikacji był obraz, na którym kolor punktów obrazował obliczoną przynależność do klas. Z przemiataniem* obrazów satelitarnych maską wiąże się problem punktów brzegowych obrazów. W naszym przypadku obraz wynikowy był mniejszy od obrazów wejściowych. Działo się tak, gdyż punkty, dla których maska wychodziła poza brzeg obrazów wejściowych, nie były uwzględniane w obrazie wyjściowym. Szerokość nieuwzględnianego marginesu równała się połowie rozmiaru maski.

Oprócz obrazów referencyjnych i parametrów klasyfikatora parametrami programu `mrf_texture` były: rozmiar kwadratowej maski (*size*), w której dokonywane są obliczenia (domyślnie 20); krok (*step*), z jakim maska przesuwana się po obrazie (domyślnie 1); nazwa obrazu testowego (domyślnie `testImage.tif`); stopień kwantyzacji histogramu chromatyczności i jego śladu (domyślnie 100); maksymalny wykładnik (*limit1*) przy współrzędnych w momentach chromatyczności (domyślnie 3); oraz nazwa obrazu wynikowego.

2.3. Eksperymenty i wyniki

Przeprowadzone eksperymenty sprowadzały się do wykonania następujących czynności:

- przygotowania obrazów referencyjnych;
- przygotowania pliku z nazwami obrazów referencyjnych i parametrami klasyfikatora najbliższego sąsiedztwa (na tym etapie wartości parametrów były dowolne, musiała jednak zgadzać się ich liczba);
- wygenerowania zestawu uczącego i estymat parametrów klasyfikatora najbliższego sąsiedztwa;

*Pojęcie przemiatania zostało wprowadzone w tym opracowaniu dla określenia całego procesu nakładania maski na poszczególne punkty obrazu i powtarzania tej czynności. Idea tego procesu jest dokładnie wyjaśniona w rozdziale 4.

- wytrenowania sieci neuronowej (jeśli segmentacja miała odbywać się za jej pomocą) lub przygotowania pliku z nazwami obrazów referencyjnych i parametrami klasyfikatora najbliższego sąsiedztwa (na tym etapie wartości parametrów musiały być równe wyliczonym estymatom);
- przeprowadzenia klasyfikacji tekstur (czyli uruchomienia programu z odpowiednią opcją).

2.3.1. Przygotowanie zestawu uczącego i estymacja parametrów

Obrazy referencyjne mogły być wygenerowane sztucznie lub też przedstawiać rzeczywiste fragmenty obrazów. W opisywanych tu eksperymentach wykorzystano rzeczywiste fragmenty obrazu wycinane ręcznie z istniejących obrazów (można to zrobić dowolnym programem obróbki obrazów rastrowych, np. `gimp`).

W zależności od liczby klas oraz przyjętego modelu zestaw uczący mógł mieć różne rozmiary wektorów wejściowych i wyjściowych. Na bieżącym etapie implementacji rozmiar wektora wejściowego jest równy $18 = 2 * 3^2$ (dla $l=3$ mamy 3^2 możliwych momentów, osobno dla histogramu chromatyczności i jego śladu). Rozmiar wektora wyjściowego jest równy liczbie wyróżnionych klas tekstur (i tym samym liczbie obrazów referencyjnych).

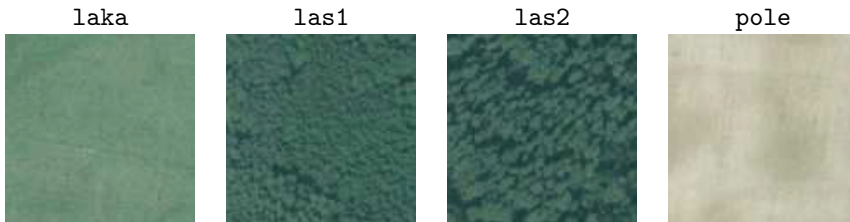
Generacja zestawu uczącego odbywała się w następujący sposób. Obraz referencyjny przemiatany był kwadratową maską o rozmiarze $size \times size$. Względne przemieszczenia maski określał parametr *step*. W każdym położeniu maski uruchamiano algorytm obliczający momenty chromatyczne. Wartości momentów definiowały wektor wejść w zestawie uczącym sieci neuronowej. n wymiarowy wektor wyjść (w przypadku n obrazów referencyjnych) tworzono, wstawiając wartość 1 na współrzędnej odpowiadającej klasie, którą dany obraz referencyjny reprezentował, oraz wartość 0 na pozostałych współrzędnych.

Ostatecznie, liczba par uczących dla jednej tekstury zależała od wartości parametrów *step*, *size* oraz wysokości i szerokości przedstawiającego ją obrazu referencyjnego.

Obliczanie momentów chromatycznych nie ma zbytniego sensu dla małych masek, toteż *size* powinno być nie mniejsze niż 10. Jednak im większy rozmiar maski, tym dłużej trwają obliczenia. Co więcej, wartości wyliczanych momentów stają się coraz większe. Na bieżącym etapie implementacji momenty chromatyczne były skalowane wartościami wynikającymi z ich rzędów (tj. dzielone przez *limit1*, podniesione do potęgi równej rzędowi momentu). Być może słuszniejsza byłaby tu normalizacja lub wykorzystanie momentów centralnych itp.

2.3.2. Wyniki segmentacji obrazu

Segmentacja tekstur polegała na uruchomieniu programu `mrf_texture` z odpowiednimi parametrami. Wynik segmentacji zdjęcia lotniczego, dokonanej na bazie dostarczonych obrazów referencyjnych, przedstawia rysunek 2.2, gdzie obrazy referencyjne przedstawia rysunek 2.1.

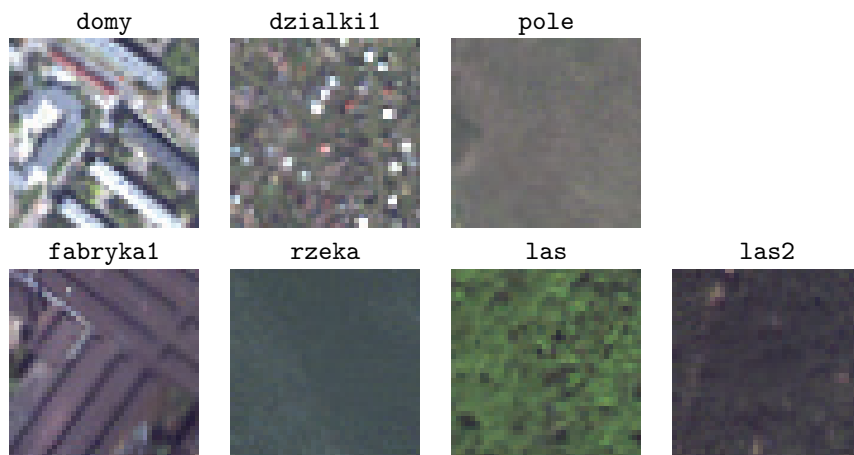


Rys. 2.1. Zdjęcie lotnicze: obrazy referencyjne.

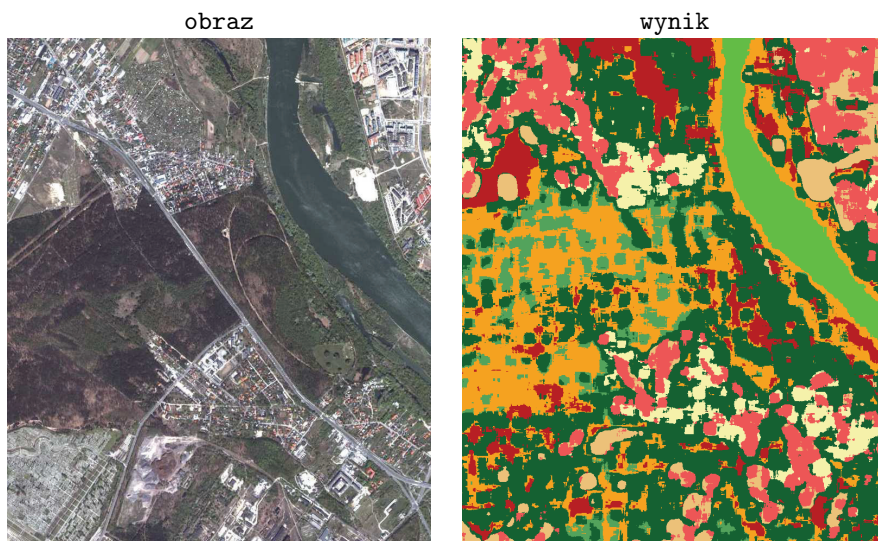


Rys. 2.2. Zdjęcie lotnicze: obraz pierwotny (l) i wynik segmentacji (p) [GUGiK].

Wynik segmentacji obrazu satelitarnego, dokonanej na bazie dostarczonych obrazów referencyjnych, przedstawia rysunek 2.4, natomiast obrazy referencyjne — rysunek 2.3.



Rys. 2.3. Obrazy satelitarne: referencyjne.



Rys. 2.4. Obraz satelitarny: obraz pierwotny (l) i wynik segmentacji (p).

3. Eksperymenty z klasyfikacją tekstur metodą losowych pól Markowa

Klasyfikacja tekstur metodą losowych pól Markowa (MRF — *Markov Random Fields*) bazuje na modelu parametrycznym, w którym występują parametryzowane funkcje losowe (Li, 1995), jakkolwiek prowadzone są badania nad nieparametrycznym modelem (Paget, 1999). Klasyfikacja tekstur polega tu na wyznaczeniu parametrów modelu zbudowanego zgodnie z tą metodą i dokonaniu klasyfikacji przy jego użyciu.

3.1. Podstawy matematyczne

3.1.1. Definicje podstawowe

Niech *pole widzenia* S będzie prostokątnym obszarem o rozmiarze $N_1 \times N_2$ zdefiniowanym jak niżej:

$$S = \{(i, j) : 1 \leq i \leq N_1, 1 \leq j \leq N_2\} \quad (3.1)$$

Elementy $s \in S$ nazwiemy pikselami.

Niech *paleta* \mathcal{L} będzie skończonym zbiorem, którego elementami są *kolory*. W ogólności dla n kolorów można zapisać:

$$\mathcal{L} = \{1, \dots, n\} \quad (3.2)$$

Funkcja $x : S \rightarrow X$ jest nazywana *obrazem*. Wartość tej funkcji dla $s \in S$ oznaczmy przez x_s (pojedynczy indeks s jest uogólnieniem podwójnego indeksowania (i, j)).

Definicja 1. *System sąsiedztwa dla S zdefiniowany jest jako:*

$$\sigma = \{\sigma_s \mid \forall s \in S\} \quad (3.3)$$

gdzie σ_s jest zbiorem pikseli sąsiadujących z pikselem s , zaś samo sąsiedztwo charakteryzuje się następującymi własnościami:

- (1) piksel s nie sąsiaduje ze sobą samym: $s \notin \sigma_s$
- (2) relacja sąsiedztwa jest wzajemna: $s \in \sigma_{s'} \Leftrightarrow s' \in \sigma_s$.

Piksele $s' \in \sigma_s$ nazywane są sąsiadami piksela s .

Dla regularnej kraty S zbiór sąsiadów piksela s zdefiniowany jest jako zbiór pikseli w promieniu r :

$$\sigma_s = \{s' \in S \mid [dist(s', s)]^2 \leq r, s' \neq s\} \quad (3.4)$$

5	4	3	4	5
4	2	1	2	4
3	1	X	1	3
4	2	1	2	4
5	4	3	4	5

Rys. 3.1. Otoczenia n -tego rzędu: pola oznaczone cyfrą 1 to otoczenie pierwszego rzędu, cyfrą 1 i 2 — otoczenie drugiego rzędu; 1,2 i 3 — otoczenie trzeciego rzędu itd.

gdzie $dist(s', s)$ oznacza odległość euklidesową pomiędzy pikselami s' i s . W przypadku dyskretnym (obrazów 2D) wyróżnia się otoczenia pierwszego, drugiego, trzeciego, ... n -tego rzędu (rys. 3.1).

Para (S, σ) tworzy graf, w którym węzłami są elementy S , zaś połączenia zdefiniowane są przez sąsiedztwo σ .

Definicja 2. *Klika c dla (S, σ) jest podzbiorem zbioru S . Klika zawierać może jeden piksel $c = \{s\}$ albo parę sąsiadujących pikseli $c = \{s, s'\}$, albo trójkę sąsiadujących pikseli $c = \{s, s', s''\}$ itd. Kolekcja klik zawierających pojedyncze piksele, pary oraz trójki oznaczona będzie, odpowiednio, przez:*

$$C_1 = \{s \mid s \in S\} \quad (3.5)$$

$$C_2 = \{\{s, s'\} \mid s' \in \sigma_s, s \in S\} \quad (3.6)$$

$$C_3 = \{\{s, s', s''\} \mid s, s', s'' \in S \text{ wszystkie są sąsiadami}\} \quad (3.7)$$

Kolekcja wszystkich klik oznaczana jest przez:

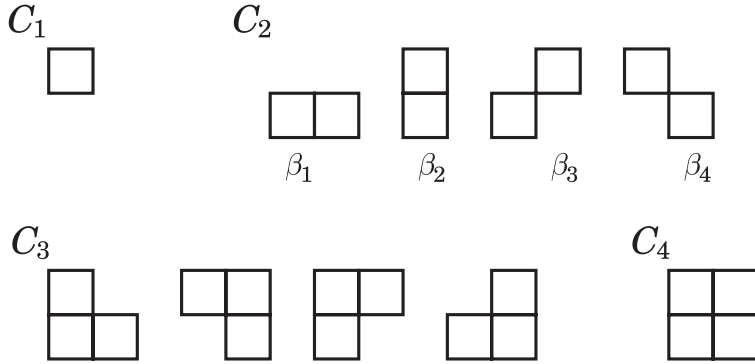
$$C = C_1 \cup C_2 \cup C_3 \dots \quad (3.8)$$

Na rysunku 3.2 przedstawiono typy klik odpowiadających otoczeniu drugiego rzędu dla regularnej kraty.

3.1.2. Pola losowe Markowa i Gibbsa

Niech $X = \{X_1, \dots, X_m\}$ (w przypadku obrazu $m = N_1 \times N_2$) będzie rodziną zmiennych losowych zdefiniowanych na zbiorze S , w której każda zmienna X_s przyjmuje wartość $x_s \in \mathcal{L}$. Rodzina X nazywana jest polem losowym.

Niech zajście zdarzenia takiego, że X_s przyjęło wartość x_s , oznaczone będzie przez $X_s = x_s$. Niech zajście łącznego zdarzenia oznaczone będzie przez $(X_1 =$



Rys. 3.2. Kliki dla otoczenia drugiego rzędu; z klikami należącymi do C_2 związane parametry β_k , $k = 1, 2, 3, 4$.

$x_1, \dots, X_m = x_m$). Dla uproszczenia zdarzenie łączne oznaczać można przez $X = x$, gdzie $x = \{x_1, \dots, x_m\}$ jest konfiguracją X odpowiadającą realizacji pola.

W przypadku dyskretnego zbioru X (ograniczona liczba kolorów) prawdopodobieństwo, że X_s przybierze wartość x_s , oznaczone będzie przez $P(X_s = x_s)$ lub, w skrócie, przez $P(x_s)$. Łączne prawdopodobieństwo $P(X = x)$ oznaczone będzie przez $P(X_1 = x_1, \dots, X_m = x_m)$ lub, w skrócie, przez $P(x)$ (w przypadku ciągłym $p(X_s = x_s)$ oraz $p(X = x)$ oznaczałyby odpowiednie funkcje gęstości prawdopodobieństwa).

Z punktu widzenia obróbki obrazów będziemy przyjmować następujące założenie:

Każdy obraz jest realizacją jakiegoś pola losowego. Każdy piksel z osobna jest realizacją zmiennej losowej, która przyjmuje wartości z przedziału zdefiniowanej paletą kolorów. $P(x_s)$ to prawdopodobieństwo wystąpienia piksela o kolorze x_s na pozycji s w obrazie.

Definicja 3. X nazywane jest losowym polem Markowa na S względem systemu sąsiedztwa σ wtedy i tylko wtedy, gdy zachodzą dwa warunki:

$$P(x) > 0, \forall x \in X \quad (\text{positivity}) \quad (3.9)$$

$$P(x_s | x_{S-\{s\}}) = P(x_s | x_{\sigma_s}) \quad (\text{markovianity}) \quad (3.10)$$

gdzie $S - \{s\}$ jest to różnica zbiorów (wszystkie piksele poza pikselem s), $x_{S-\{s\}}$ to wartości (kolory) wszystkich pikseli poza pikselem s , oraz

$$x_{\sigma_s} = \{x_{s'} \mid s' \in \sigma_s\} \quad (3.11)$$

jest zbiorem zawierającym wartości (kolory) pikseli sąsiadujących z pikselem s .

Markovianity pokazuje lokalną charakterystykę pola X . Z własności tej wynika, że o kolorze danego piksela decyduje kolor jego sąsiadów. Z polem Markowa wiążą się jeszcze własności: homogeniczności i izotropiczności.

Definicja 4. Zbiór zmiennych losowych X nazywany jest polem losowym Gibbsa (GRF — Gibbs Random Field) na S względem σ wtedy i tylko wtedy, gdy jego łączna dystrybucja ma postać:

$$P(x) = \frac{1}{Z} e^{-\frac{1}{T}U(x)}, \quad (3.12)$$

gdzie

$$Z = \sum_{x \in X} e^{-\frac{1}{T}U(x)} \quad (3.13)$$

jest czynnikiem normalizującym, zapewniającym $\sum_{x \in X} p(x) = 1$, natomiast T jest stałą nazywaną temperaturą (można przyjąć $T = 1$). Z kolei:

$$U(x) = \sum_{c \in C} V_c(x) \quad (3.14)$$

jest funkcją energii będącą sumą potencjałów $V_c(x)$ wszystkich możliwych klik należących do C , przy czym wartość $V_c(x)$ zależy od lokalnej konfiguracji klikli c .

GRF jest polem homogenicznym, jeśli $V_c(x)$ nie zależy od względnego położenia klikli c w S . Pole to jest izotropiczne, jeśli V_c nie zależy od orientacji klikli c .

Uwaga: MRF charakteryzowane jest przez swoją lokalną własność (*markovianity*), podczas gdy GRF charakteryzowane jest przez swoją globalną własność (dystrybucję Gibbsa). Teoria Hammersleya-Clifforda (Hammersley and Clifford, 1971) ustanowiła równoważność pomiędzy tymi dwoma typami własności. Głosi ona, że X jest MRF na S względem σ wtedy i tylko wtedy, gdy X jest GRF na S względem σ .

Prawdopodobieństwo warunkowe $P(x_s | x_{S-\{s\}})$ (wystąpienia piksela o kolorze x_s przy ustalonym jego otoczeniu) wyraża się wzorem:

$$P(x_s | x_{S-\{s\}}) = \frac{e^{-\sum_{c \in \mathcal{A}} V_c(x_s)}}{\sum_{x'_s} e^{-\sum_{c \in \mathcal{A}} V_c(x'_s)}} \quad (3.15)$$

gdzie x'_s to wszystkie możliwe kombinacje, jakie może przyjąć x_s , \mathcal{A} to zbiór klik zawierających piksel s . Prawdopodobieństwo to zależy tylko od potencjału klik zawierających piksel s , a więc od kolorów pikseli z jego lokalnego otoczenia. Wynik ten dowodzi, że GRF jest MRF.

3.2. Model opisu tekstury

Wybór postaci i parametrów funkcji potencjału jest podstawowym zadaniem w modelowaniu MRF. Formy funkcji potencjału jednoznacznie definiują postać

dystrybucji Gibbsa. W naszym przypadku przyjęliśmy model określany w literaturze nazwą: *multi-level logistic (MLL) model* (Derin and Elliott, 1987), *generalized Ising model* (Geman and Geman, 1984) w lekko zmodyfikowanej postaci. W oryginalnym modelu potencjał klikli zawierającej przynajmniej dwa piksele wyraża się wzorem:

$$V_c(x) = \begin{cases} \zeta_c & \text{jeśli wszystkie piksele w klicie } c \text{ mają ten sam kolor} \\ -\zeta_c & \text{w przeciwnym przypadku} \end{cases} \quad (3.16)$$

gdzie ζ_c jest potencjałem klikli typu c .

W naszym przypadku przyjęto założenie, że w modelu występują tylko potencjały klikli o liczności 2 dla otoczenia rzędu drugiego, a potencjały innych klikli równają się zeru. Niech więc z klikami należącymi do C_2 skojarzone będą parametry jak na rysunku 3.2. Potencjał klikli typu k należącej do C_2 określa teraz wzór:

$$V_2(x_s, x_{s'}) = \begin{cases} -\beta_k & \text{jeśli oba piksele w klicie typu } k \text{ mają ten sam kolor} \\ \beta_k & \text{w przeciwnym przypadku} \end{cases} \quad (3.17)$$

gdzie β_k , $k = 1, 2, 3, 4$ to parametr związany z typem klikli.

Niech $\bar{\beta} = (\beta_1, \dots, \beta_4)$ będzie wektorem parametrów. Potencjał klikli należącej do C_2 można zapisać jako $V_2(x_s, x_{s'}, \bar{\beta})$, podkreślając zależność od parametrów.

Ostatecznie prawdopodobieństwo warunkowe $P(x_s | x_{S-\{s\}})$ dla tak przyjętego modelu wyraża się wzorem:

$$P(x_s | x_{S-\{s\}}) = \frac{e^{-\sum_{s'} V_2(x_s, x_{s'})}}{\sum_{x_{s'}} e^{-\sum_{s'} V_2(x_{s'}, x_{s'})}} \quad (3.18)$$

gdzie $\sum_{s'}$ oznacza sumowanie po wszystkich sąsiadach piksela s tworzących z nim kliklię należącą do C_2 . Dla podwójnego indeksowania wzór ten można rozpiszać następująco:

$$P(x_{i,j} | x_{S-\{(i,j)\}}) = \frac{e^{-L(x_{i,j})}}{\sum_{x'_{i,j}} e^{-L(x'_{i,j})}} \quad (3.19)$$

gdzie

$$L(a_{i,j}) = V_2(a_{i,j}, a_{i-1,j}) + V_2(a_{i,j}, a_{i+1,j}) + V_2(a_{i,j}, a_{i,j-1}) + V_2(a_{i,j}, a_{i,j+1}) + V_2(a_{i,j}, a_{i-1,j-1}) + V_2(a_{i,j}, a_{i+1,j+1}) + V_2(a_{i,j}, a_{i+1,j-1}) + V_2(a_{i,j}, a_{i-1,j+1}) \quad (3.20)$$

Dla przykładu, jeśli we wszystkich klikach występują piksele jednakowego koloru, to L wyraża się przez:

$$L = 2\beta_1 + 2\beta_2 + 2\beta_3 + 2\beta_4 \quad (3.21)$$

jeśli zaś kolory pikseli w każdej klicie są różne, to:

$$L = -2\beta_1 - 2\beta_2 - 2\beta_3 - 2\beta_4 \quad (3.22)$$

3.3. Synteza tekstury

Generowanie tekstury odpowiadającej realizacji dystrybucji Gibbsa $P(x) = \frac{1}{Z}e^{-U(x)}$ odbywa się przez próbkowanie dystrybucji. Często wykorzystywane w tym celu są dwa algorytmy: *Metropolis sampler* (Metropolis et al., 1953) oraz *Gibbs sampler* (Geman and Geman, 1984).

3.3.1. Metropolis sampler

Algorytm 1 (*Metropolis sampler*) jest algorytmem iteracyjnym, w którego iteracjach dla wybranego piksela losowany jest nowy kolor (z rozkładem równomiernym), przy czym akceptowany jest on z prawdopodobieństwem p .

Algorytm 1.

Początek.

- (1) Zainicjuj kolory pikseli danego obrazu x losowo wybranymi kolorami z palety \mathcal{L} ;
- (2) dla piksela $s \in S$ obrazu rób, co następuje:
 - (2.1) niech y będzie kopią obrazu x ;
 - (2.2) niech x_s będzie kolorem wylosowanym z palety \mathcal{L} ;
 - (2.3) oblicz $p = \min \{ 1, P(y)/P(x) \}$,
gdzie P jest daną dystrybucją Gibbsa (z przyjętymi dla syntezy tekstury parametrami);
 - (2.4) z prawdopodobieństwem p dokonaj przypisania $x = y$;
- (3) powtórz (2) N razy.

Koniec.

3.3.2. Gibbs sampler

Algorytm 2 (*Gibbs sampler*) jest algorytmem iteracyjnym, w którego iteracjach dla wybranego piksela proponowane są wszystkie możliwe kolory i akceptowane są one z prawdopodobieństwem obliczanym z dystrybucji Gibbsa w lokalnym otoczeniu.

Algorytm 2.

Początek.

- (1) Zainicjuj kolory pikseli danego obrazu x losowo wybranymi kolorami z palety \mathcal{L} ;
- (2) dla piksela $s \in S$ obrazu rób, co następuje:
 - (2.1) oblicz $p_l = P(x_s = l \mid x_{\sigma_s})$
dla wszystkich kolorów $l \in \mathcal{L}$ przy znanych kolorach sąsiadów x_{σ_s} ;
 - (2.2) z prawdopodobieństwem p_l dokonaj przypisania $x_s = l$
- (3) powtórz (2) N razy.

Koniec.

3.4. Segmentacja tekstur

Segmentację tekstury na bazie modelu losowych pól Markowa można przeprowadzić w dwojaki sposób:

- Wyznaczając wartości parametrów modelu w lokalnym otoczeniu (zdefiniowanym położeniem maski) i traktując je jako cechy definiujące teksturę. Podejście to pozwala rozpoznawać pojawienie się na obrazie nowych klas. Złożoność obliczeniowa tego sposobu jest bardzo wysoka.
- Obliczając wartości funkcji wiarygodności w lokalnym otoczeniu (zdefiniowanym położeniem maski) dla zestawów parametrów modelu odpowiadającym referencyjnym teksturom. W podejściu tym parametry, dla których funkcja wiarygodności osiąga wartość największą, definiują wynik klasyfikacji. Złożoność obliczeniowa tego sposobu jest stosunkowo mała.

W obu przypadkach podstawą do segmentacji były wyliczane wartości parametrów modelu (pierwszy sposób wymaga obliczania parametrów w każdym położeniu maski na obrazie klasyfikowanym, drugi wymaga obliczenia wartości parametrów tylko na etapie budowania modelu na bazie obrazów referencyjnych tekstur). Do wyliczania parametrów modelu wykorzystano poniższe algorytmy (Yu and Cheng, 2003):

Algorytm 3.

(0) Inicjalizacja.

Niech $t = 0$. Wybierz początkowy zestaw parametrów $\bar{\beta}_0$.

Określ wartość η oraz maksymalną liczbę iteracji N_1 .

(1) Jeśli $t > N_1$ to koniec; w przeciwnym wypadku wykonuj, co następuje:

(2) Wylosuj $\nu = \text{rand}(0,1)$.

Jeśli $\nu \leq \eta$, to

a) oblicz gradient funkcji wiarygodności dla $\bar{\beta} = \bar{\beta}_t$:

$$dPL = \left. \frac{\partial PL(x|\bar{\beta})}{\partial \bar{\beta}} \right|_{\bar{\beta}=\bar{\beta}_t} \quad (3.23)$$

b) wyznacz wartość $\bar{\beta}'$ na pozytywnym kierunku tego gradientu:

$$\bar{\beta}' = \bar{\beta}_t + \frac{dPL}{\|dPL\|} \cdot |\text{normal}(0,1)| \quad (3.24)$$

gdzie $\text{normal}(\mu, \sigma)$ to generator liczb losowych o rozkładzie normalnym ze średnią μ i wariancją σ ;

c) oblicz $\alpha(\bar{\beta}', \bar{\beta}_t)$:

$$\alpha(\bar{\beta}', \bar{\beta}_t) = \min \left\{ 1, e^{PL(x|\bar{\beta}') - PL(x|\bar{\beta}_t)} \frac{1 - \eta}{2\eta(\sqrt{2\pi^3})} \right\} \quad (3.25)$$

w przeciwnym wypadku ($\nu > \eta$):

- a) niech $\bar{\beta}'$ będzie wartością otrzymaną z generatora liczb losowych o rozkładzie normalnym:

$$\bar{\beta}' = \text{normal}(\bar{\beta}_t, I_4) \quad (3.26)$$

gdzie I_4 to macierz jednostkowa.

- b) oblicz $\alpha(\bar{\beta}', \bar{\beta}_t)$:

$$\alpha(\bar{\beta}', \bar{\beta}_t) = \min \left\{ 1, e^{PL(x|\bar{\beta}') - PL(x|\bar{\beta}_t)} \right\} \quad (3.27)$$

- (3) Jeśli $\text{rand}(0,1) < \alpha(\bar{\beta}', \bar{\beta}_t)$,
to $\bar{\beta}_{t+1} = (\bar{\beta}')$;
w przeciwnym razie $\bar{\beta}_{t+1} = (\bar{\beta}_t)$.
(4) $t = t+1$, skocz do (1).

Algorytm 4.

- (0) Inicjalizacja. Niech $t=0$. Wybierz początkowy zestaw parametrów $\bar{\beta}_0$. Określ temperaturę początkową T_0 , maksymalną liczbę iteracji N_2 oraz wyznacz wartość γ z równania $T_0 \gamma^{N_2} = 1$.
(1) Niech $\bar{\beta}'$ będzie wartością otrzymaną z generatora liczb losowych o rozkładzie normalnym:

$$\bar{\beta}' = \text{normal}(\bar{\beta}_t, I_4) \quad (3.28)$$

gdzie I_4 to macierz jednostkowa.

- (2) Oblicz $\alpha(\bar{\beta}', \bar{\beta}_t)$:

$$\alpha(\bar{\beta}', \bar{\beta}_t) = \min \left\{ 1, e^{\frac{PL(x|\bar{\beta}') - PL(x|\bar{\beta}_t)}{T_t}} \right\} \quad (3.29)$$

- (3) Jeśli $\text{rand}(0,1) < \alpha(\bar{\beta}', \bar{\beta}_t)$, to $\bar{\beta}_{t+1} = \bar{\beta}'$;
w przeciwnym razie $\bar{\beta}_{t+1} = \bar{\beta}_t$.
(4) $T_{t+1} = \gamma T_t$, $t = t+1$, skocz do (1).

Algorytm 5.

- (0) Inicjalizacja. Wybierz początkowy zestaw parametrów $\bar{\beta}_0$.
(1) Uruchom Algorytm 4 z wybranymi parametrami $\bar{\beta}_0$.
(2) Zapisz wynik z Algorytmu 4 jako początkowy zestaw parametrów $\bar{\beta}_0$ dla Algorytmu 3.
(3) Uruchom Algorytm 3.
(4) Powtórz M razy ($M \leq N_1$) kroki (0), (1), (2), (3).
Zapisz wyniki z kroku 3 jako $\tilde{\beta}_i$, $i=1, \dots, M$.
Estymatę parametrów oblicz ze wzoru:

$$\tilde{E}(\bar{\beta}|x) = \frac{1}{M} \sum_{i=1}^M \tilde{\beta}_i \quad (3.30)$$

3.5. Szczegóły implementacji

Generacja sztucznych tekstur, przygotowanie zestawów uczących i segmentacja tekstur to kolejne etapy przetwarzania danych, które zostały zaimplementowane w języku C++ z wykorzystaniem biblioteki `OpenCV`. Wynikiem implementacji jest program o nazwie `mrf_texture`. Do celów wizualizacji wyników segmentacji zaimplementowano w języku C++, w oparciu o bibliotekę `Qt`, dodatkowy program `testQt-02`.

Do generacji sztucznych tekstur wykorzystano Algorytm 2. Krok (2) zaimplementowano, wyznaczając dystrybuantę rozkładu Gibbsa, obliczając następnie jej odwrotność dla wartości wylosowanej z przedziału $(0, 1)$ z równomiernym rozkładem prawdopodobieństwa. Algorytm uruchamiano pięć razy, przy czym każde uruchomienie wiązało się z wyznaczeniem kolorów dla wszystkich pikseli obrazu.

Krok (2) Algorytmu 2.

- (1) oblicz $Z = \sum_{l \in \mathcal{L}} U(l)$, gdzie $l = 1, \dots, n$
- (2) oblicz $Q_k = \sum_{l=1}^k U(l)/Z$, gdzie $k = 1, \dots, n$
- (3) wylosuj $p = \text{rand}(0, 1)$
- (4) for ($k = 1$; $k < n$; $k++$)
 if ($p < Q_k$) break;
- (5) nowy kolor równa się k

3.6. Eksperymenty

Eksperymenty, mówiąc ogólnie, polegały na wykonaniu poniższych czynności:

- przygotowaniu obrazów referencyjnych,
- wygenerowaniu zestawu uczącego dla sieci neuronowej,
- wytrenowaniu sieci neuronowej (przy założeniu, że segmentacja będzie wykonywana za jej pomocą) lub obliczeniu estymat parametrów (gdyby segmentacja miała być realizowana metodą funkcji największej wiarygodności),
- przeprowadzeniu klasyfikacji tekstur (czyli uruchomieniu programu z odpowiednią opcją).

3.6.1. Przygotowanie obrazów referencyjnych

Obrazy referencyjne mogły być wygenerowane sztucznie lub też przedstawiać rzeczywiste tekstury. W pierwszym przypadku przygotowanie tych obrazów polegało na utworzeniu plików z teksturami wygenerowanymi za pomocą programu `mrf_texture`. W drugim przypadku przygotowanie plików z teksturami polegało na ręcznym wycinaniu jednolitych pod względem tekstury fragmentów z istniejących obrazów. Operację wycinania udostępnia szereg programów do obróbki obrazów rastrowych, jak np. program `gimp` lub narzędzia pakietu `ImageMagick`.

3.6.2. Generacja zestawu uczącego dla sieci neuronowej i estymacja parametrów

W zależności od liczby klas oraz przyjętego modelu zestaw uczący mógł mieć różne rozmiary wektorów wejściowych i wyjściowych. Na bieżącym etapie implementacji rozmiar wektora wejściowego równy jest 4, gdyż taka jest liczba parametrów w przyjętym modelu pól losowych. Rozmiar wektora wyjściowego odpowiada zaś liczbie wyróżnionych klas tekstur (liczbie obrazów referencyjnych).

Generacja zestawu uczącego odbywała się w następujący sposób. Obraz referencyjny, zawierający teksturę wzorcową, przemiatany był kwadratową maską o rozmiarze $size \times size$. Względne przemieszczenia maski określał parametr *step*. W każdym położeniu maski uruchamiany był algorytm estymujący 4 parametry modelu. Liczbę uruchomień algorytmu w danym położeniu maski określał parametr *trials_no*. Ostateczny wynik, tj. uśrednione wartości każdego z czterech parametrów uzyskane w *trials_no* próbach dla jednego położenia maski, definiował wektor wejść w zestawie uczącym sieci neuronowej. Ten n -wymiarowy wektor wyjść (w przypadku n obrazów referencyjnych) tworzono, wstawiając wartość 1 na współrzędnej odpowiadającej analizowanej klasie tekstury oraz wartość 0 na pozostałych współrzędnych. Ostatecznie, liczba par uczących dla jednej tekstury zależała od wartości parametrów *step*, *size* oraz wysokości i szerokości obrazu referencyjnego reprezentującego teksturę.

Ze względu na to, że przyjęty model bazuje na otoczeniu drugiego rzędu, rozmiar maski musi spełniać zależność $size \geq 3$. Im większy rozmiar maski, tym lepiej estymują się parametry modelu. Jednak im większa maska, tym większe są problemy z właściwą estymacją parametrów na granicach obszarów.

3.6.3. Trenowanie sieci

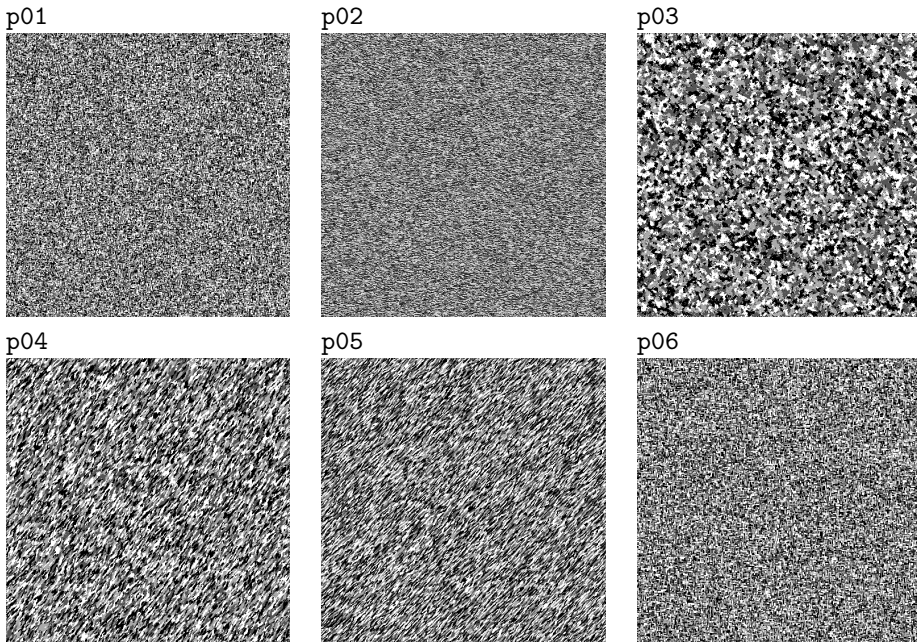
Trenowanie sieci odbywało się za pomocą programowego narzędzia SNNS. Na początek tworzono sieć typu *feed-forward* o architekturze odpowiadającej wygenerowanemu zestawowi uczącemu. Tak więc liczby wejść i wyjść sieci zgadzały się z rozmiarami wektorów w parach uczących. Liczba warstw ukrytych oraz ich liczność mogła być dowolna, jednak w naszym przypadku przyjęto 2 warstwy ukryte po 8 neuronów każda. Połączenia neuronów pomiędzy warstwami sieci były typu każdy z każdym.

Przed przystąpieniem do uczenia sieci jej wagi inicjalizowane były losowo. Samo uczenie odbywało się zgodnie z wybranym algorytmem (**Backpropagation** lub **Rprop**). Nauczona sieć zapisywana była następnie do pliku pod nazwą odpowiadającą jej architekturze, np. `i4h8h8o6.net`. Używając narzędzia `snn2c`, tworzono plik nagłówkowy i plik źródłowy, zawierający implementację wytrenowanej sieci w języku C, a następnie linkowano z programem `mrf_texture`.

3.6.4. Segmentacja sztucznych tekstur

Segmentacja tekstur polegała na uruchomieniu programu `mrf_texture` z odpowiednimi parametrami. Sztuczne tekstury, które służyły później jako obrazy referencyjne (rys. 3.3), wygenerowano dla następującego zestawu parametrów:

obraz	β_1	β_2	β_3	β_4
p01	1.0	1.0	-0.5	-0.5
p02	1.0	-0.8	0.5	-0.5
p03	0.3	0.3	0.3	0.3
p04	0.5	1.0	-0.5	0.7
p05	1.0	1.0	-1.0	1.0
p06	2.0	2.0	-1.0	-1.0



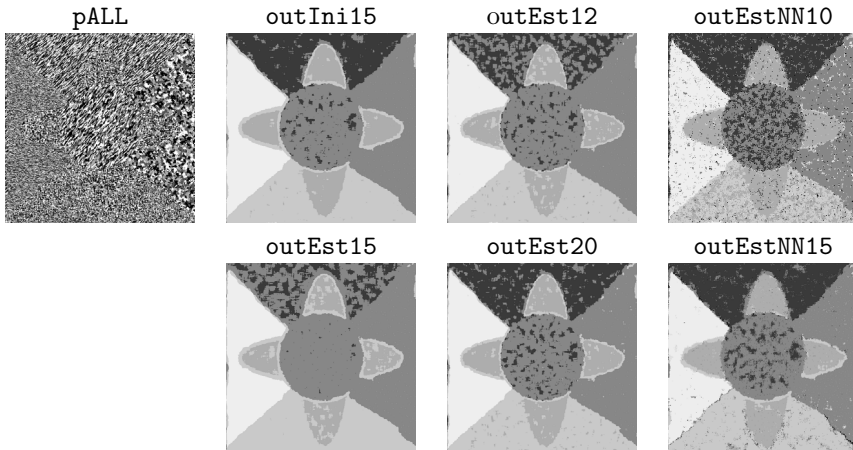
Rys. 3.3. Sztuczne tekstury.

Uśrednione, estymowane wartości parametrów, obliczone na podstawie sztucznych tekstur, były następujące:

obraz	β_1	β_2	β_3	β_4
p01	1.08717	1.09241	-0.565433	-0.566961
p02	1.04465	-0.899957	0.558655	-0.54535
p03	0.320296	0.314226	0.288998	0.281211
p04	0.74194	1.30031	-0.843298	0.704515
p05	1.53794	1.48114	-1.6005	1.53289
p06	2.4197	2.45401	-1.24345	-1.30294

Klasyfikacji poddawano obraz pALL (rys. 3.4), zawierający fragmenty wszystkich sześciu sztucznych tekstur. Podczas klasyfikacji używano masek o różnych rozmiarach. Poniższe zestawienie przedstawia nazwy wygenerowanych obrazów wynikowych, otrzymanych metodą funkcji największej wiarygodności przy wartości $skok=1$ dla parametrów oryginalnych (dla których wygenerowane zostały sztuczne tekstury) oraz dla parametrów estymowanych (wyznaczonych na bazie referencyjnych sztucznych tekstur).

parametry		rozmiar maski
oryginalne	estymowane	
outIni15	outEst15	15×15
outIni12	outEst12	12×12
outIni10	outEst10	10×10



Rys. 3.4. Wybrane wyniki segmentacji sztucznych tekstur.

Do klasyfikacji siecią neuronową (pierwszy sposób klasyfikacji) utworzono sieć typu *feed-forward* o czterech wejściach, dwóch warstwach ukrytych po 8 neuronów każda i sześciu wyjściach. Aby wytrenować sieć, przygotowano zestaw uczący, analizując w tym celu obrazy referencyjne. Trenowanie sieci odbywało się algorytmem *Rprop*. Wyniki klasyfikacji siecią neuronową przedstawiają obrazy, których nazwy, wartość parametru *skok* oraz rozmiar maski przedstawione są w poniższym zestawieniu:

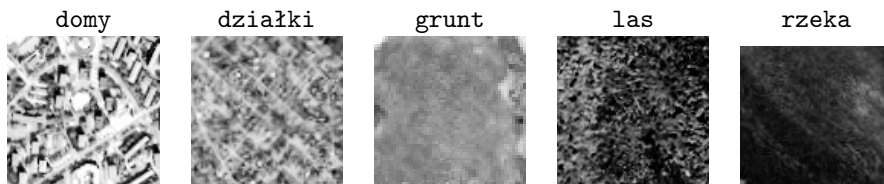
	<i>skok</i>	rozmiar maski
outEstNN10	2	10×10
outEstNN15	1	15×15

Wybrane wyniki segmentacji pokazano na rysunku 3.4. Dowodzą one ogólnej poprawności podejścia, tzn. tekstury były ogólnie poprawnie rozpoznane, z dokładnością zależną od konkretnych zastosowanych parametrów.

3.6.5. Segmentacja tekstur naturalnych

Eksperymenty z naturalnymi teksturami przeprowadzono w podobny sposób jak te z teksturami sztucznymi. Jedyną różnicą polegała na tym, że obrazy referencyjne nie były wygenerowane sztucznie (dla znanych parametrów), lecz były teksturami naturalnymi.

Obrazem wejściowym do eksperymentów był wielospektralny obraz satelitarny o rozdzielczości 1 m, dla którego utworzono barwną kompozycję RGB dla pasm, odpowiednio: 1, 2 i 3. Przestrzeń barw otrzymanego obrazu przekształcono następnie do przestrzeni HSV i wyselekcjonowano kanał V (wartość). Otrzymany obraz (256 odcieni szarości) stanowił bazę do przygotowania właściwych danych do eksperymentu. Z obrazu tego wycięto fragment testowy *wartosc*, który podlegał klasyfikacji, oraz fragmenty wzorcowe: *domy*, *działki*, *grunt*, *las*, *rzeka*, które pełniły rolę obrazów referencyjnych zawierających jednorodną teksturę (rys. 3.5).

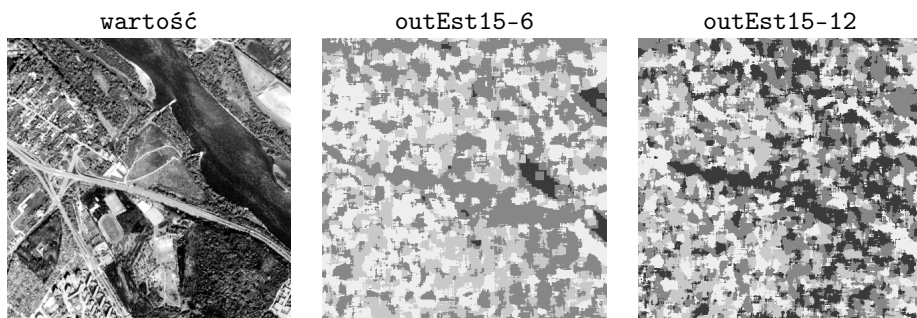


Rys. 3.5. Wzorce tekstur naturalnych.

We wszystkich eksperymentach z naturalnymi teksturami używano maski o rozmiarze 15×15 . Uśrednione estymowane wartości parametrów, uzyskane dla poszczególnych referencyjnych tekstur, były następujące:

obraz	β_1	β_2	β_3	β_4
domy	0.494702	0.550263	0.0208361	0.111291
działki	0.401439	0.423862	0.120034	0.142854
grunt	0.867212	0.813467	0.58215	-0.174253
las	0.589072	0.393893	0.0747596	0.1255
rzeka	1.16445	1.95361	2.16164	-0.222192

Klasyfikację tekstur naturalnych przeprowadzono obydwoma sposobami opisanymi w podrozdziale 3.4. W przypadku sieci neuronowych zastosowano sieć typu *feed-forward* o identycznej architekturze jak sieć w eksperymencie ze sztucznymi teksturami. Część uzyskanych wyników przedstawia rysunek 3.6 (pozostałe wyniki nie odbiegały znacząco od wyników zaprezentowanych).



Rys. 3.6. Wyniki segmentacji sztucznych tekstur. Obraz po lewej stronie to obraz oryginalny. Obraz środkowy przedstawia wyniki uzyskane przy kwantyzacji kolorów do 6 poziomów. Obraz prawy — wyniki uzyskane przy kwantyzacji kolorów do 12 poziomów.

Na podstawie przedstawionych na rysunku 3.6 wyników można wyciągnąć wniosek, że przyjęty model pola losowego nie nadaje się do reprezentowania naturalnych tekstur. Najwyraźniej, pomimo iż model ten dobrze sprawdza się dla tekstur sztucznych, to zawodzi dla tekstur naturalnych.

4. Detekcja i klasyfikacja obszarów na zdjęciach lotniczych za pomocą sieci neuronowych

Definicja klas pokrycia terenu identyfikowanych na zdjęciach lotniczych w większym stopniu koresponduje z gospodarczym aspektem cech wspólnych obiektów niż jednolitością i podobieństwem tych obiektów na zdjęciach. Ta cecha obrazów lotniczych przysparza dużych kłopotów w poprawnej klasyfikacji jednorodnych pod względem tematycznym obiektów a różnych, często znacznie, pod względem odbicia światła zarejestrowanego na materiale światłoczułym. Przykładem może być klasa *budynki*, która — w zależności od zastosowanego pokrycia dachu, stopnia nasłonecznienia i kąta padania promieni słonecznych — często wykazuje dużą rozpiętość cech spektralnych. Inny problem dotyczy podobieństwa w odbiciu światła od obiektów różnych ze względu na aspekt użytkowania a wykonanych z podobnych materiałów, jak np. budynki kryte papą i drogi asfaltowe. Ponadto, w odróżnieniu od obrazów satelitarnych, fotogrametryczne zdjęcia lotnicze wykonywane są zazwyczaj tylko w widzialnym zakresie spektrum, przez co możliwość ich analizy poprzez kombinacje kanałów spektralnych jest wykluczona lub co najmniej mocno ograniczona. Wszystkie te cechy zdjęć lotniczych powodują konieczność odejścia od wyłącznej klasyfikacji na podstawie cech pojedynczego piksela — takich jak odbicie światła — i włączenia do analizy informacji z treści otoczenia klasyfikowanego punktu obrazu. Generalnie istnieją dwa podejścia do analizy informacji kontekstowej elementów obrazu. Jedno oparte jest na kompleksowej segmentacji obrazu, drugie — na statystycznej analizie sąsiedztwa pikseli w oknie przeglądającym. W drugiej grupie metod szeroko rozpowszechnione są algorytmy oparte na analizie tekstury obrazu.

Zastosowanie sieci neuronowych do rozwiązywania problemów w kartografii i systemach GIS znane jest już od kilkunastu lat (Meng, 1993; Müller, 1992; Openshaw et al., 1991). W tej stosunkowo nowej i wciąż rozwijającej się dziedzinie sieci neuronowe są stosowane do: klasyfikacji tematycznej (semantycznej) atrybutów obiektów, klasyfikacji obiektów przestrzennych, agregacji i generalizacji itp. Przykładem może tu być zastosowanie sieci neuronowych do rozpoznawania wzorców i klasyfikacji ze względu na sposób użytkowania na obrazach rastrowych, np. obrazach satelitarnych, (Meng, 1993)). Początkowo zainteresowanie sieciami neuronowymi wiązało się z problemem klasyfikacji *per pixel*, gdzie z zastosowaniem sieci neuronowych wiązano nadzieję na zwiększenie dokładności przestrzennej ograniczonej przez rozdzielczość dostępnych danych. Późniejsze prace koncentrowały się na klasyfikacji *per parcel* (lub *per field*), w której łączono techniki używane w klasyfikacji *per pixel* z dodatkowym przetwarzaniem danych. Często istotą dodatkowego przetwarzania było wykorzystanie statystycznych właściwości badanych obszarów (Berberoglu et al., 2000; German and Gahegan, 1996; Liu et al., 2002; Miller et al., 1995; Tatem et al., 2001; Venkatesh and Raja, 2003).

4.1. Rozpoznawanie tekstur

Tekstura jest cechą, która pozwala rozróżniać obszary w skali makroskopowej. W niniejszym podrozdziale przedstawione zostaną dwie pokrewne metody ekstrahujące cechy, na bazie których można dokonywać segmentacji tekstur. Szersze omówienie zagadnienia rozpoznawania i syntezy tekstur można znaleźć w pracach (Haralick, 1986; Tuceryan and Anil, 1998)).

4.1.1. Wariogram

Wariogram określa się jako semiwariancję w funkcji przesunięcia (Atkinson and Emery, 1999; Atkinson and Lewis, 2000; Berberoglu et al., 2000; Carr, 1996, 1997):

$$\gamma(\mathbf{h}) = \frac{1}{2}E [(Z(\mathbf{x}) - Z(\mathbf{x} + \mathbf{h}))^2] \quad (4.1)$$

którą można estymować wariogramem przykładu $\gamma(\mathbf{h})$ dla $p(\mathbf{h})$ par obserwacji, $\{z(\mathbf{x}_l), z(\mathbf{x}_l + \mathbf{h}), l = 1, 2, \dots, p(\mathbf{h})\}$ jak niżej:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2}p(\mathbf{h}) \sum_{l=1}^{p(\mathbf{h})} (z(\mathbf{x}_l) - z(\mathbf{x}_l + \mathbf{h}))^2 \quad (4.2)$$

Wariogramu, podobnie jak innych cech obrazujących lokalną statystykę, można użyć do klasyfikacji obszarów na obrazach satelitarnych (i zdjęciach lotniczych). Więcej informacji na temat różnych postaci wariogramów można znaleźć w pracy (Chica-Olmo and Abarca-Hernández, 2000). Jest tam porównanie rezultatów klasyfikacji dla różnych pasm spektralnych Landsat 5 oraz różnych wariogramów.

4.1.2. Metoda macierzy GLCM w analizie tekstur

Metoda macierzy GLCM (ang. *Gray Level Co-occurrence Matrix*) pozwala wyznaczyć parametry tekstury, a co za tym idzie, dokonać segmentacji obrazu. Zalicza się ona do szerokiej gamy metod badających relację sąsiedztwa pikseli na analizowanym obrazie.

Analiza relacji sąsiedztwa w metodzie GLCM polega na zbudowaniu pewnej macierzy, której wartości elementów odzwierciedlają liczbę wystąpień par pikseli o tych samych kolorach w zadanej odległości od siebie, na zadanym kierunku i w zadanym obszarze, na całym obrazie lub też na jego fragmencie. Stąd też pochodzi nazwa metody: macierz wspólnych wystąpień szarych punktów. Macierz sąsiedztwa przedstawić można wzorem:

$$V_{i,\alpha}(i, j) = |\{(r, s), (t, v) : I(r, s) = i, I(t, v) = j\}| \quad (4.3)$$

gdzie $i, j = 0, \dots, (N - 1)$ to poziomy szarości punktów na obrazie odległych o l na kierunku α , N — liczba poziomów szarości, $I(x, y)$ — wartość koloru punktu

o współrzędnych (x, y) na obrazie, $(t, v) = (r + l \cos \alpha, s + l \sin \alpha)$. Stąd widać, że macierz sąsiedztwa V jest macierzą kwadratową, a jej rozmiar zdefiniowany jest liczbą poziomów szarości obrazu. Kolumny macierzy GLCM odpowiadają wartościom poziomów szarości punktów sąsiednich, natomiast wiersze — wartościom poziomów szarości punktów odniesienia. Stąd wartości elementów w macierzy określają, ile razy piksel o danym odcieniu sąsiadował z pikselem o innym odcieniu w analizowanym obszarze.

Na bazie macierzy sąsiedztwa wyznaczane są pewne cechy statystyczne analizowanego obszaru. Macierz sąsiedztwa poddawana jest symetryzacji (dzięki symetryczności wystarczy analizować jedynie piksele pod jednym z kątów w danym kierunku tzn. dla kąta 90° i 270° macierze sąsiedztwa będą identyczne):

$$\bar{V}_{l,\alpha} = \frac{V_{l,\alpha} + V_{l,\alpha}^T}{2} \quad (4.4)$$

i normalizacji liczbą punktów analizowanego obszaru (aby wartości elementów mieściły się w granicach $(0, 1)$):

$$P_{i,j} = \frac{\bar{V}_{l,\alpha}(i,j)}{\sum_{i,j=0}^{N-1} \bar{V}_{l,\alpha}(i,j)} \quad (4.5)$$

Symetryczna i znormalizowana macierz $P_{i,j}$ stanowi wyjście do obliczenia cech statystycznych. Z jej definicji wynika, że oprócz parametrów l i α istotnym jest również parametr określający wielkość analizowanego obszaru. Stąd, aby zmniejszyć ilość generowanych danych (statystycznych cech), macierze sąsiedztwa wyznaczone dla różnych parametrów często się uśrednia. Metoda macierzy GLCM wymaga dużych nakładów obliczeniowych. Aby je zmniejszyć, można przeskalować występujące na obrazie poziomy szarości, co skutkuje pomniejszeniem ich liczby. Tym samym zmniejszy się rozmiar macierzy sąsiedztwa.

W eksperymentach opisanych w dalszej części tego opracowania posłużono się parametrami statystycznymi zdefiniowanymi w następującej tabeli.

kontrast	$\sum_{i,j=0}^{N-1} P_{i,j} (i-j)^2$
odmienność	$\sum_{i,j=0}^{N-1} P_{i,j} i-j $
podobieństwo	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$
max	$Max(P_{i,j})$
ASM (<i>Angular Second Moment</i>) drugi moment kątowy	$\sum_{i,j=0}^{N-1} P_{i,j}^2$
energia	\sqrt{ASM}
entropia	$\sum_{i,j=0}^{N-1} P_{i,j} (-\ln P_{i,j})$

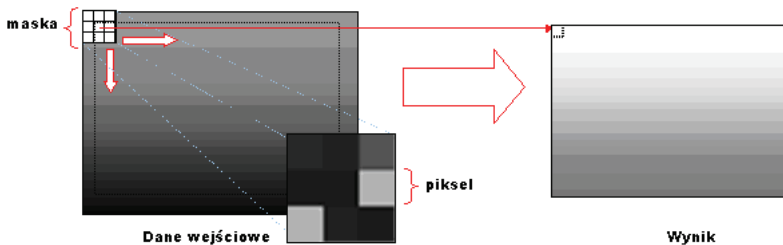
4.2. Użycie sieci neuronowych z maską

4.2.1. Założenia eksperymentu

Aby przeprowadzić klasyfikację zdjęcia lotniczego, obejmującą rozpatrywany zestaw klas użytkowania, w projekcie zastosowano sztuczne sieci neuronowe z nadzorowaną metodą ich uczenia. Eksperymenty wykonano na kolorowych ortoobrazach lotniczych o terenowej rozdzielczości piksela około 0.5 metra. Przeprowadzono rozpoznanie i klasyfikację następujących klas:

- Budynki
- Drogi
- Drzewa

Tematyka klas podyktowana była zapotrzebowaniem na tego typu analizy w związku z budową w Polsce regionalnych systemów informacji przestrzennej. Proces obliczeniowy obejmował testowanie jakości klasyfikacji ze względu na skalę klasyfikowanego zdjęcia i rozmiar otoczenia klasyfikowanego piksela (liczbę pikseli podawanych na wejście sieci neuronowej). Podzbiór pikseli stanowiących treść informacji kontekstowej zdefiniowany został jako kwadratowe okno przesuwne zwane maską. Za piksel klasyfikowany przyjęto centralny piksel w masce (w przypadku nieparzystego rozmiaru) lub pierwszy z lewej z dwóch centralnych pikseli (w przypadku parzystego rozmiaru maski). Proces przemiataania klasyfikowanego zdjęcia maską, a więc kolejność tworzenia wektorów cech, odbywał się od lewego górnego rogu w kierunku prawym (wiersz po wierszu) ze skokiem jednego piksela. Zilustrowany na rysunku 4.1 proces przetwarzania powoduje zmniejszenie obszaru wynikowego klasyfikacji w stosunku do obrazu wejściowego o szerokość $\frac{1}{2}$ rozmiaru maski z każdej strony z dokładnością do 1 piksela.

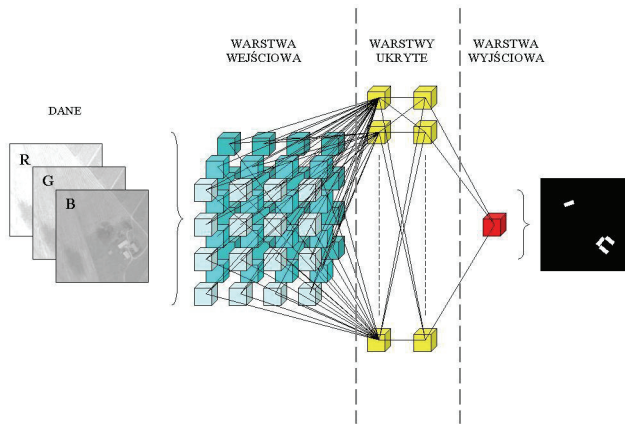


Rys. 4.1. Schemat przetwarzania maską.

Zdjęcie lotnicze analizowane było w dwóch rozdzielczościach terenowych: około 0.5×0.5 m (oryginalnej) i otrzymanej przez przeskalowanie oryginalnej — około 1×1 m (zredukowanej). Testowano następujące wielkości masek:




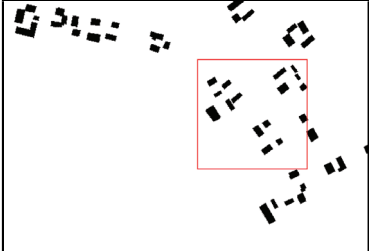
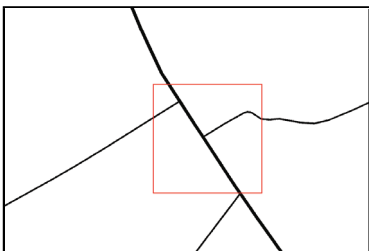
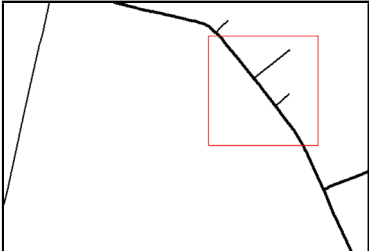


- Dla rozdzielczości 0.5 m : 15×15 , 19×19 , 31×31 , 39×39 , 47×47
- Dla rozdzielczości 1 m : 7×7 , 9×9 , 15×15 , 19×19 , 23×23

Rozmiary masek dobierano tak, aby w przybliżeniu obejmowały taki sam obszar w terenie dla obu rozpatrywanych rozdzielczości. We wszystkich przypadkach zdjęcia rozbijane były na kanały RGB i do wektora wejściowego sieci wartości poszczególnych kanałów podawano równocześnie. Sieć neuronowa wyposażona była w dwie warstwy ukryte po 10 neuronów każda. Liczba wejść do sieci neuronowej determinowana jest przez rozmiar maski oraz liczbę kanałów przestrzeni barw użytych do klasyfikacji. Uczono rozpoznawać tylko jedną klasę, toteż tylko jeden neuron wyjściowy był niezbędny. Schemat użytej sieci neuronowej ilustruje rysunek 4.2.



Rys. 4.2. Schemat wielowarstwowej sieci neuronowej z jednym neuronem wyjściowym.

Aby przeprowadzić proces uczenia sieci metodą nadzorowaną, niezbędne jest przygotowanie zestawu uczącego, w skład którego prócz zestawów n danych wejściowych U zwanych wektorem cech $u_i(n)$ musi wchodzić zestaw oczekiwanej odpowiedzi sieci $d(n)$, dzięki któremu błąd odpowiedzi sieci (różnica pomiędzy wartością oczekiwaną a obliczoną: $\epsilon(n) = d(n) - y(n)$) może być obliczony i wstecznie propagowany w celu korekcji wag $w_i(n)$. Wektor cech U składał się z trzech znormalizowanych zestawów wartości pikseli (natężenia światła dla każdego kanału RGB), pobranych z obrazu o 8-bitowej rozdzielczości kolorów. Wzorec klasyfikacji, z którego generowano wartości $d(n)$, uzyskano z Topograficznej Bazy Danych (TBD) i stanowił on binarny obraz rastrowy. Podobnie skonstruowany został drugi, różny od zestawu uczącego, zestaw danych zwany zestawem testowym. Sklasyfikowany zestaw testowy porównywany był do jego wzorca klasyfikacji z TBD w celu obliczenia miar jakości.

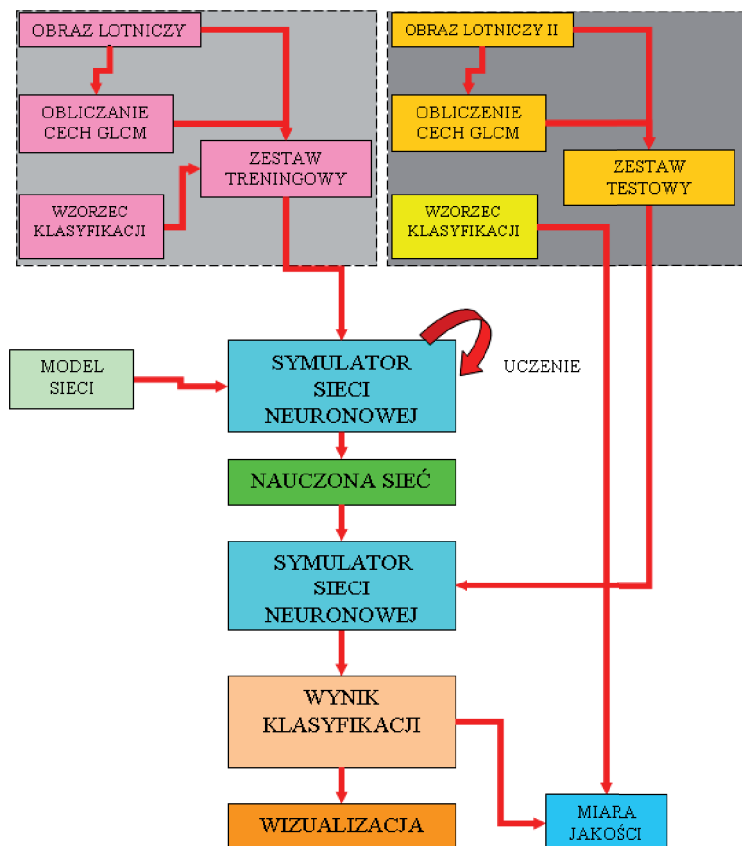
		Zestaw uczący	Zestaw testowy
Zdjęcia lotnicze			
Klasy wyznaczone arbitralnie	Budynki		
	Drogi		
	Lasy		

Rys. 4.3. Zbiór uczący i zbiór testowy (zdjęcia lotnicze) razem z obrazami reprezentującymi klasy: Budynki, Drogi, Lasy. Czerwone kwadraty zaznaczone na obrazach odpowiadają obszarom o wymiarze ok. 300×300 m w terenie [GUGiK].

Zestawy uczące oraz testowe wraz ze wzorcami szukanych klas przedstawiono na rysunku 4.3. W czasie eksperymentów przetwarzane były całe zdjęcia i dla całych zdjęć obliczane były numeryczne miary oceny poprawności klasyfikacji. Aby

zachować przejrzystość prezentowanych wyników, wizualne efekty przeprowadzonej klasyfikacji zaprezentowano dalej tylko dla zaznaczonych obszarów.

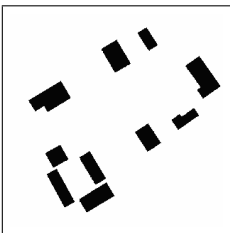
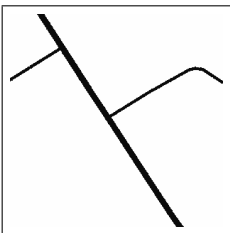

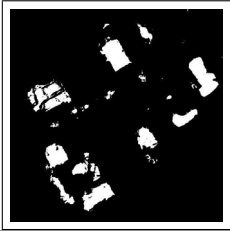
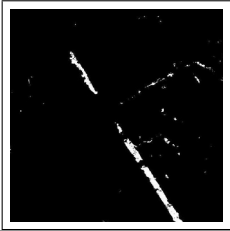
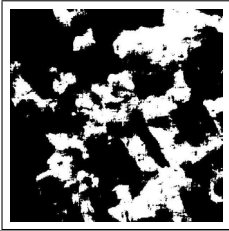

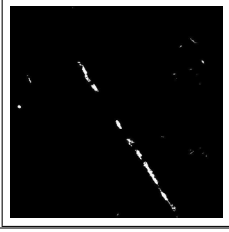
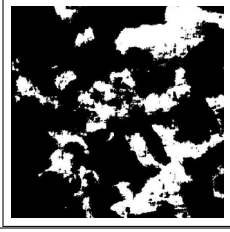
Proces obliczeniowy sieci neuronowych oparto o symulator SNNS (*Stuttgart Neural Network Simulator*). W uczeniu z nauczycielem wykorzystano metodę wstecznej propagacji błędów. Schemat przetwarzania zdjęć ilustruje rysunek 4.4.



Rys. 4.4. Schemat przetwarzania zdjęć.

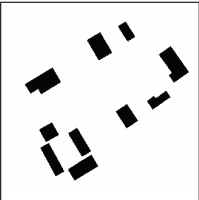
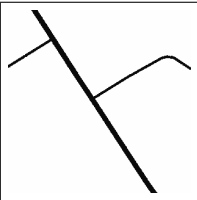

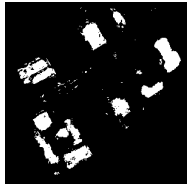
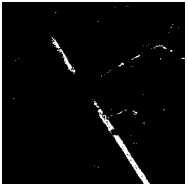
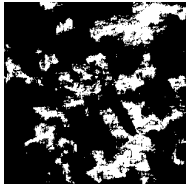



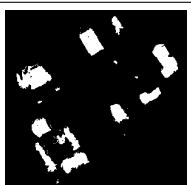
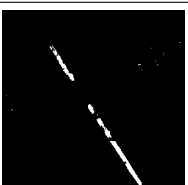
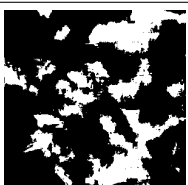


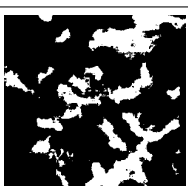
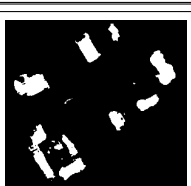
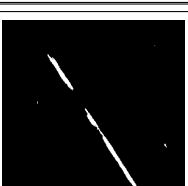
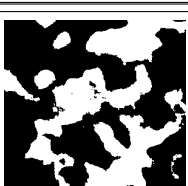
4.2.2. Eksperymenty

Wizualne rezultaty klasyfikacji zdjęć lotniczych o półmetrowej rozdzielczości terenowej przedstawiono na rysunku 4.5. Wyniki obliczono dla masek o rozmiarach 15×15 oraz 19×19 pikseli, tylko dla zbioru uczącego.

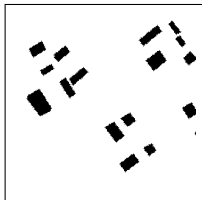
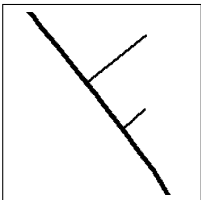

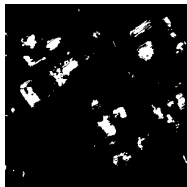
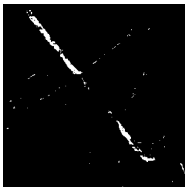
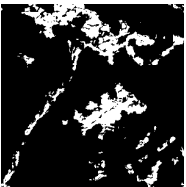

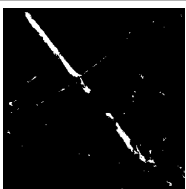
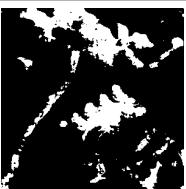
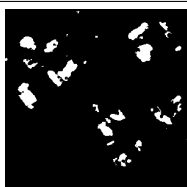
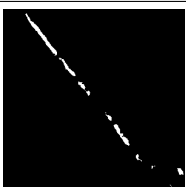

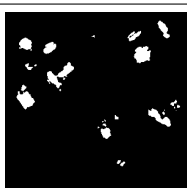
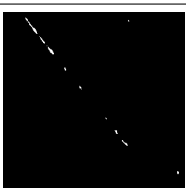

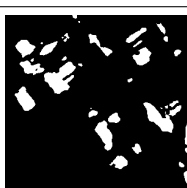


Typ		Budynki	Drogi	Lasy
Obrazy referencyjne				
Wyniki klasyfikacji	15×15			
	19×19			

Rys. 4.5. Wyniki klasyfikacji dla zbioru uczącego dla różnych rozmiarów masek dla zdjęć o rozdzielczości terenowej 0.5 m.

Rysunki 4.6 i 4.7 przedstawiają wyniki eksperymentów dla obrazów o rozdzielczości metrowej otrzymane z zastosowaniem masek o rozmiarach: 7×7 , 9×9 , 15×15 , 19×19 , i 23×23 pikseli. Zarówno zdjęcie uczące, jak i testowe miały rozmiar terenowy ok. 1000×700 m, co w przełożeniu na rozmiary obrazów odpowiadało wielkości 2000×1400 pikseli dla rozdzielczości półmetrowej oraz 1000×700 pikseli dla zdjęć o metrowej rozdzielczości terenowej. Uzyskane wyniki wskazują na sensowność dalszego zwiększania rozmiarów masek. Niestety, złożoność obliczeniowa eksperymentów z maską o rozmiarach 31×31 pikseli i większych dla półmetrowej rozdzielczości przekraczały możliwości obliczeniowe zastosowanego sprzętu i nie zostały przeprowadzone.

Typ	Budynki	Drogi	Lasy	
Obrazy referencyjne				
Wyniki klasyfikacji	7×7			
	9×9			
	15×15			
	19×19			
	23×23			

Rys. 4.6. Wyniki klasyfikacji dla zbioru uczącego dla różnych rozmiarów masek dla zdjęć o rozdzielczości terenowej 1 m.

Typ	Budynki	Drogi	Lasy	
Obrazy referencyjne				
Wyniki klasyfikacji	7×7			
	9×9			
	15×15			
	19×19			
	23×23			

Rys. 4.7. Wyniki klasyfikacji dla zbioru testowego dla różnych rozmiarów masek dla zdjęć o rozdzielczości terenowej 1 m.

4.2.2.1. Ocena jakości klasyfikacji

Aby dokonać ilościowej oceny klasyfikacji, obliczono miary jakości klasyfikacji zaproponowane przez autorów w rozdziale 1.6. Zestawienie tych miar dla rozdzielczości 1 m zawierają poniższe tabele.

Ocena wyników klasyfikacji dla zestawu uczącego								
Typ	Maska	κ_w	u_1	u_2	p_1	p_2	d	uDW
Budynki	7×7	0.9778	0.8152	0.9861	0.7481	0.9906	0.9778	0.1987
	9×9	0.9776	0.8406	0.9841	0.7131	0.9925	0.9777	0.1752
	15×15	0.9785	0.8538	0.9846	0.7256	0.9930	0.9786	0.1617
	19×19	0.9786	0.8229	0.9870	0.7709	0.9905	0.9787	0.1901
	23×23	0.9766	0.8313	0.9836	0.7129	0.9917	0.9765	0.1851
Drogi	7×7	0.9857	0.7983	0.9868	0.2655	0.9988	0.9856	0.2149
	9×9	0.9875	0.8006	0.9893	0.4115	0.9981	0.9876	0.2101
	15×15	0.9851	0.9141	0.9854	0.1944	0.9997	0.9851	0.1006
	19×19	0.9840	0.7103	0.9854	0.2035	0.9985	0.9840	0.3043
	23×23	0.9850	0.8991	0.9853	0.1975	0.9996	0.9849	0.1156
Lasy	7×7	0.9230	0.6914	0.9484	0.5913	0.9660	0.9233	0.3602
	9×9	0.9289	0.6945	0.9587	0.6792	0.9614	0.9291	0.3468
	15×15	0.9352	0.7269	0.9622	0.7099	0.9652	0.9357	0.3108
	19×19	0.9353	0.7740	0.9516	0.6224	0.9761	0.9350	0.2745
	23×23	0.9224	0.6246	0.9766	0.8314	0.9338	0.9218	0.3988

Ocena wyników klasyfikacji dla zestawu testowego								
Typ	Maska	κ_w	u_1	u_2	p_1	p_2	d	uDW
Budynki	7×x7	0.9716	0.9863	0.5534	0.9843	0.5865	0.9715	0.4604
	9×9	0.9726	0.9900	0.4831	0.9819	0.6298	0.9727	0.5270
	15×15	0.9736	0.9883	0.5587	0.9843	0.6303	0.9735	0.4529
	19×19	0.9726	0.9902	0.4838	0.9817	0.6397	0.9727	0.5260
	23×23	0.9638	0.9734	0.6998	0.9890	0.4863	0.9638	0.3269
Drogi	7×7	0.9837	0.9984	0.1166	0.9852	0.5459	0.9836	0.8850
	9×9	0.9837	0.9973	0.1820	0.9863	0.5363	0.9837	0.8207
	15×15	0.9850	0.9996	0.1128	0.9852	0.8431	0.9849	0.8876
	19×19	0.9841	0.9999	0.0378	0.9840	0.9375	0.9840	0.9622
	23×23	0.9844	0.9989	0.1310	0.9856	0.6599	0.9845	0.8701
Lasy	7×7	0.9464	0.9711	0.5446	0.9723	0.5336	0.9467	0.4843
	9×9	0.9458	0.9633	0.6616	0.9791	0.5230	0.9460	0.3751
	15×15	0.9430	0.9604	0.6607	0.9789	0.5051	0.9432	0.3789
	19×19	0.9395	0.9527	0.7193	0.9823	0.4822	0.9392	0.3281
	23×23	0.9441	0.9622	0.6566	0.9786	0.5164	0.9446	0.3812

4.2.2.2. Wnioski z eksperymentu

Opisany eksperyment miał na celu wykazanie możliwości identyfikacji na wysokorozdzielczych zdjęciach fotogrametrycznych obiektów, takich jak budynki, drogi czy drzewa. Już wizualna analiza wyników przekonuje, że przy odpowiednim doborze parametrów procesu klasyfikacji możliwe jest uzyskanie zadowalających wyników. Jakość klasyfikacji zależy od charakteru obiektów klasyfikowanych. Najlepsze wyniki uzyskano dla klasy Drzewa, co jest zrozumiałe ze względu na jej cechy odbicia światła. Zarejestrowane na materiale światłoczułym zadrzewienia wyraźnie kontrastują barwą od pozostałych form pokrycia terenu. Oczywiście kolor pojedynczego piksela nie jest wystarczającą przesłanką do zaliczenia go do danej klasy. Sprogowanie obrazu byłoby niewystarczające, czego dowodem jest silna korelacja pomiędzy wielkością maski a jakością klasyfikacji. Zatem informacja o otaczających pikselach jest niezbędna. Większe problemy sprawiła poprawna klasyfikacja klas Budynki i Drogi. W tym przypadku zróżnicowanie materiału z jakiego wykonano te obiekty, czyli niejednorodność klasy względem odbicia światła, pogarszało wyniki. Optymalny rozmiar maski zależy od skali zdjęcia i rodzaju klasy. W tabeli przedstawiono uzyskane w eksperymencie optymalne wymiary masek.

Rozdzielczość [m]	Klasa	Rozmiar maski
0.5	Budynki	15×15
0.5	Drogi	15×15
0.5	Drzewa	15×15
1	Budynki	19×19
1	Drogi	9×9
1	Drzewa	15×15

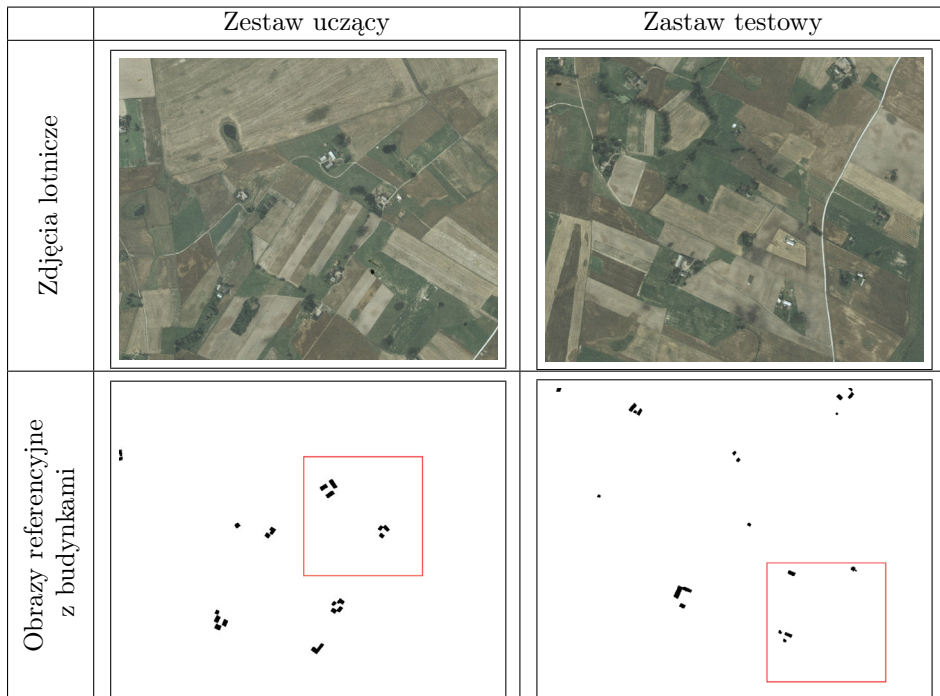
Otrzymane rezultaty, a także złożoność obliczeniowa problemu sugerują, iż stosowanie zdjęć o niższej, 1-metrowej rozdzielczości nie wpływa znacząco na jakość klasyfikacji, natomiast w dużym stopniu skraca czas obliczeń.

4.3. Użycie sieci neuronowych z algorytmem identyfikacji tekstur GLCM

W celu poprawy jakości klasyfikacji zastosowano inne podejście do ekstrakcji informacji kontekstowej. Otoczenia pikseli zastąpiono cechami opisującymi teksturę w pewnym obszarze. Tu także zastosowano metodę przemiatania obrazu maską. Obliczenia wykonano przy użyciu metody identyfikacji tekstur GLCM. Otrzymany zestaw parametrów GLCM, łącznie z wartością jasności klasyfikowanego piksela, stanowił wektor cech podawany sieci neuronowej. Sieć, stanowiąca klasyfikator, operowała znacznie zredukowanym, w porównaniu do poprzedniego eksperymentu, wektorem cech dla każdego piksela. W ten sposób znacznie skrócono czas obliczeń. Aby ułatwić ocenę przydatności obliczanych cech do klasyfikacji, zastosowano ich wizualizację poprzez zrzutowanie na 8-bitową skalę odcieni szarości.

4.3.1. Metoda GLCM w identyfikacji budynków na zdjęciach lotniczych

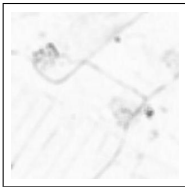
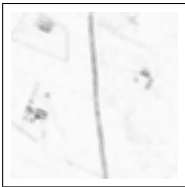

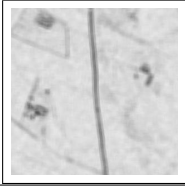
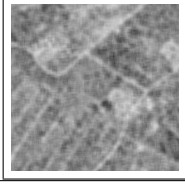
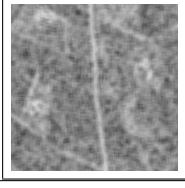
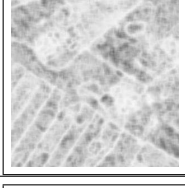
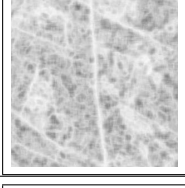

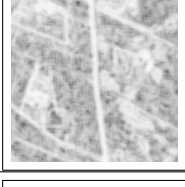

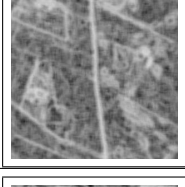


Na jakość klasyfikacji pokrycia na zdjęciach lotniczych duży wpływ ma wielkość okna maski GLCM, która zależy od klasyfikowanego typu obiektów oraz skali zdjęcia. Wyznaczenie optymalnej wielkości maski dla poszczególnych zjawisk i skal powinno być przedmiotem dalszych badań. Ułatwieniem interpretacji uzyskiwanych parametrów GLCM jest możliwość ich prezentacji w formie graficznej poprzez liniowe przekształcenie w przestrzeń 8-bitową.



Rys. 4.8. Zbiór uczący i zbiór testowy (zdjęcia lotnicze) razem z obrazami reprezentującymi klasę Budynki [GUGiK].

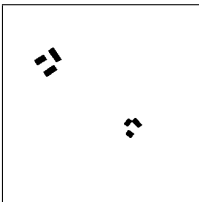
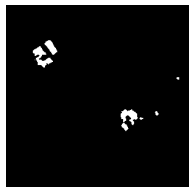
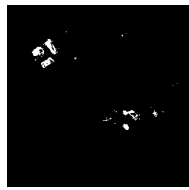
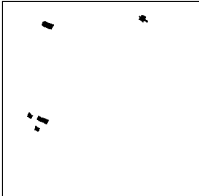
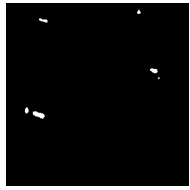
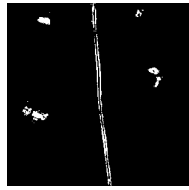
Wykonano eksperyment wykrywania budynków na zdjęciu lotniczym. W skład wektora uczącego wchodziły trzy kanały RGB oraz uśrednione parametry GLCM. Zdjęcie uczące i weryfikujące (rozdzielczość terenowa około 1 metra) oraz ich wzorce przedstawia rysunek 4.8.

Obliczono parametry GLCM dla każdego z kierunków i je uśredniono. Zastosowano maskę 10×10 pikseli. Obliczono parametry statystyczne i po ich wizualizacji i ocenie wzrokowej wykluczono z wektora uczącego parametr *Max*. Wizualizację uzyskanych parametrów statystycznych przedstawia rysunek 4.9.

	uczenie	testowanie
kontrast		
zróżnicowanie		
podobieństwo		
max		
ASM		
energia		
entropia		

Rys. 4.9. Wizualizacja parametrów GLCM dla zdjęcia lotniczego z budynkami.

W celu oceny wpływu włączenia parametrów GLCM do wektora uczącego zawierającego już kanały RGB wykonano uczenie sieci dla takich samych danych wejściowych, lecz pozbawionych tych parametrów (w wektorze uczącym pozostawiono jedynie kanały RGB). Wszystkie parametry uczenia i wizualizacji dla obu przypadków były takie same. Wynik uczenia sieci (dla zdjęcia uczącego) oraz weryfikacji na drugim zdjęciu dla danych uwzględniających parametry GLCM i dla danych pozbawionych tych wielkości przedstawia rysunek 4.10.

Typ: Budynki	Obraz referencyjny	Wyniki klasyfikacji	
		GLCM+RGB	RGB
zestaw uczący			
zestaw testowy			

Rys. 4.10. Wyniki klasyfikacji dla zbioru uczącego i testowego przy uwzględnieniu różnych cech (GLCM+RGB, RGB) dla obrazów o rozdzielczości terenowej 1 m.

Dokładności klasyfikacji osiągnięte w eksperymencie przedstawia tabela zawierająca wartości miar oceny wyników klasyfikacji dla zbioru uczącego i testowego, przy uwzględnieniu różnych cech (I — GLCM+RGB, II — RGB), dla obrazów o rozdzielczości terenowej 1 m. Indeksy w macierzy niezgodności A : i — klasa otrzymana, j — klasa dostarczona.

Ocena wyników klasyfikacji dla typu: Budynki								
dane	ce- chy	κ_w	u_1	u_2	p_1	p_2	d	uDW
zestaw uczący	I	0.9966	0.5532	0.9986	0.6353	0.9980	0.9966	0.4482
	II	0.9974	0.5339	0.9995	0.8202	0.9979	0.9974	0.4666
zestaw testowy	I	0.9972	0.2345	0.9997	0.7218	0.9975	0.9973	0.7657
	II	0.9932	0.4707	0.9949	0.2318	0.9983	0.9932	0.5344

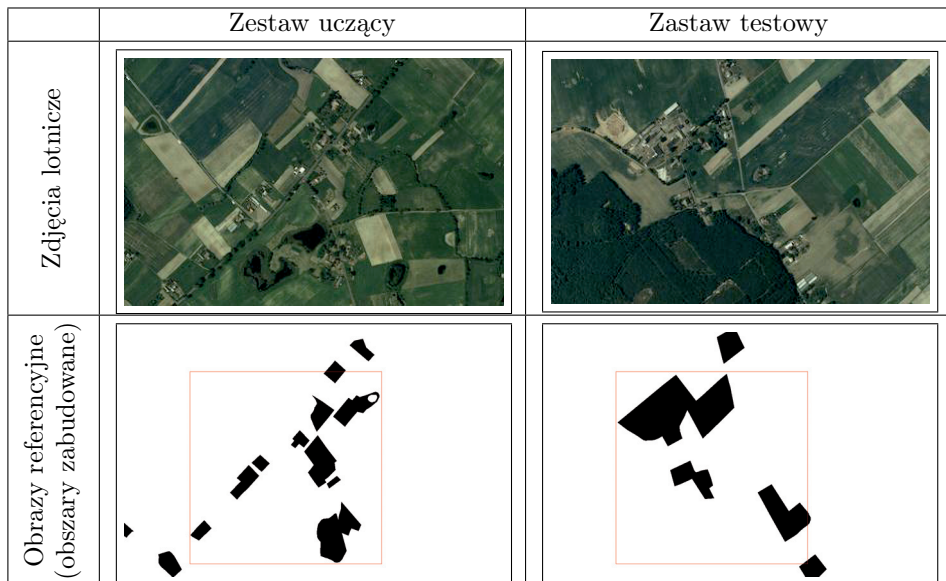
4.3.1.1. Wnioski

Włączenie do wektora uczącego parametrów GLCM zmniejsza liczbę błędnych przyporządkowań pikseli nie należących do szukanej klasy. Zmniejsza także liczbę prawidłowo rozpoznanych pikseli. Mając jednak na uwadze fakt wskazania miejsc, które należą bez wątpliwości do klasy Budynki, zastosowanie rozszerzonego wektora polepsza rozpoznanie (nie wystąpiło np. błędne zakwalifikowanie drogi).

4.3.2. Klasyfikacja obszarów zabudowanych na zdjęciach lotniczych z zastosowaniem metody GLCM

Ten eksperyment polegał na przeprowadzeniu uczenia rozpoznawania obszarów zabudowanych na zdjęciach fotogrametrycznych za pomocą sieci neuronowych. W wektorze cech klasyfikowanych obszarów uwzględniano wybrane parametry GLCM oraz wartości RGB.

Rozdzielczość terenowa zarówno zestawu uczącego, jak i testowego wynosiła około 1 m. Oba zestawy (rys. 4.11) obejmowały obszar o powierzchni około 97 ha. W celu polepszenia przejrzystości w dalszej części zaprezentowano jedynie fragmenty wyników obejmujące obszary o wymiarach około 600×600 m, zaznaczone czerwonym kwadratem.



Rys. 4.11. Zestaw uczący oraz zestaw testowy wykorzystany w eksperymencie z klasyfikacją obszarów zabudowanych z użyciem cech GLCM oraz RGB [GUGiK].

Obrazy w skali szarości poddano analizie tekstur. Obliczono następujące parametry GLCM przy założeniu maski o rozmiarze 30×30 pikseli i kwantyzacji do 32 odcieni szarości:

- kontrast
- odmienność
- podobieństwo
- maksymalne $P_{i,j}$ w masce
- ASM (Angular Second Moment)
- energia
- entropia

Zwizualizowane parametry GLCM dla fragmentu zdjęcia z zestawu uczącego przedstawiono na rysunku 4.12. W pierwszej kolumnie umieszczono cechy uśrednione, w pozostałych cechy obliczone dla każdego z możliwych kierunków.

Otrzymane parametry GLCM oraz wzorzec dla obrazka uczącego posłużyły do nauczania sieci neuronowej stanowiącej klasyfikator. Zbudowano sieci: z pojedynczą warstwą ukrytą zawierającą 9 neuronów oraz jednym wyjściem. Warstwa wejściowa miała wielkość od 5 do 23 neuronów w zależności od rozmiaru wektora uczącego. We wszystkich prócz pierwszego eksperymentu użyto zredukowany zestaw cech GLCM:

- energia
- entropia
- kontrast
- podobieństwo
- zróżnicowanie

Sieć uczono przez 1000 cykli. Wykonano klasyfikację przy następujących założeniach:

1. klasyfikacja wszystkich, uśrednionych (7) cech GLCM wraz z wartościami RGB;
2. klasyfikacja nieuśrednionych, zredukowanych cech GLCM (20) wraz z wartościami RGB;
3. klasyfikacja nieuśrednionych, zredukowanych cech GLCM (20) bez RGB;
4. klasyfikacja uśrednionych, zredukowanych cech GLCM (5) wraz z wartościami RGB;
5. klasyfikacja uśrednionych, zredukowanych cech GLCM (5) bez RGB.

	0	45	90	135	180
kontrast					
rozróżnicowanie					
podobieństwo					
max					
ASM					
energia					
entropia					

Rys. 4.12. Wizualizacja parametrów GLCM dla obrazka uczącego.

Wyniki klasyfikacji dla obrazka uczącego wg powyższych oznaczeń ilustruje rysunek 4.13.

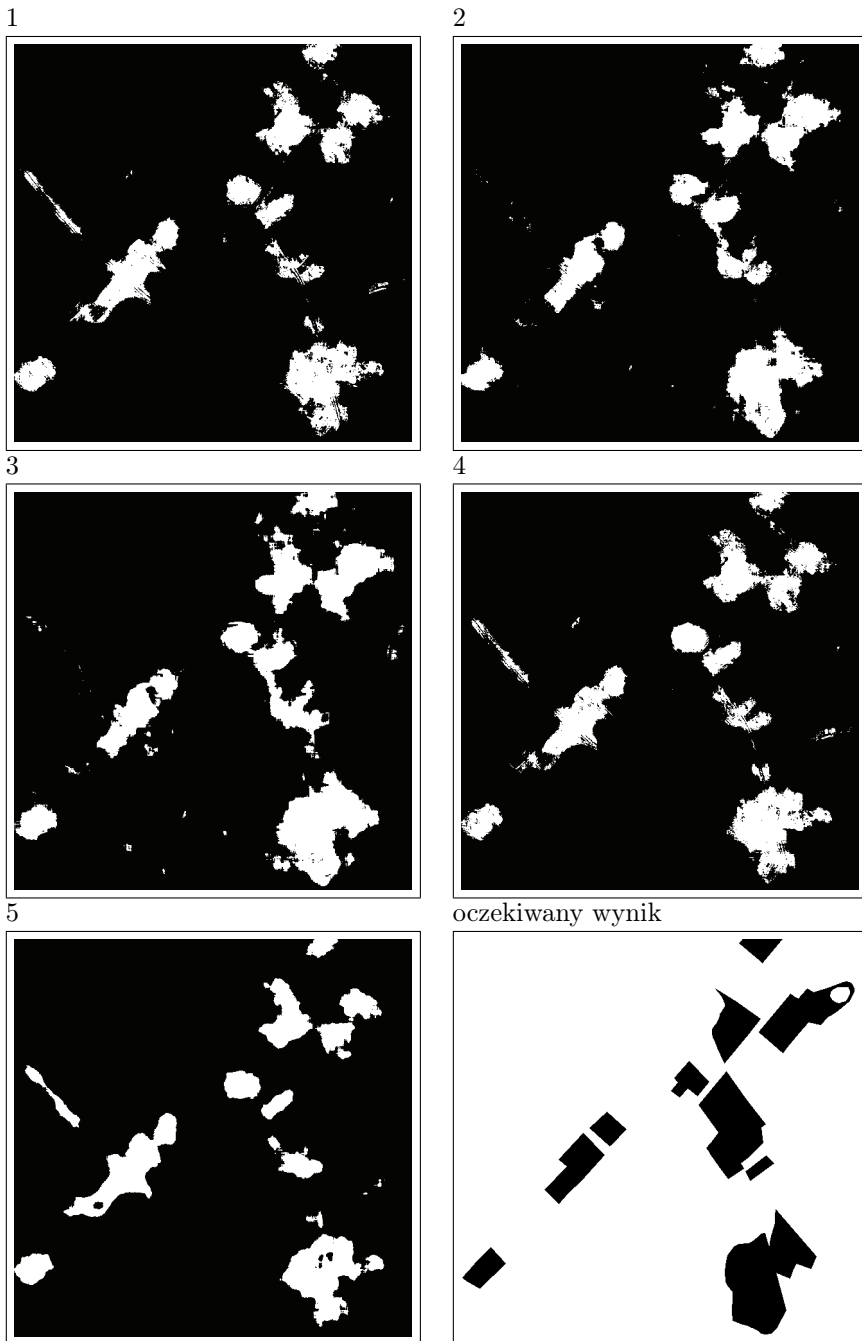
W celu weryfikacji nauczonych sieci wykonano klasyfikację obrazka testowego i otrzymano wyniki:

1. klasyfikacja wszystkich, uśrednionych (7) cech GLCM wraz z wartościami RGB;
2. klasyfikacja nieuśrednionych, zredukowanych cech GLCM (20) wraz z wartościami RGB;
3. klasyfikacja nieuśrednionych, zredukowanych cech GLCM (20) bez RGB;
4. klasyfikacja uśrednionych, zredukowanych cech GLCM (5) wraz z wartościami RGB;
5. klasyfikacja uśrednionych, zredukowanych cech GLCM (5) bez RGB.

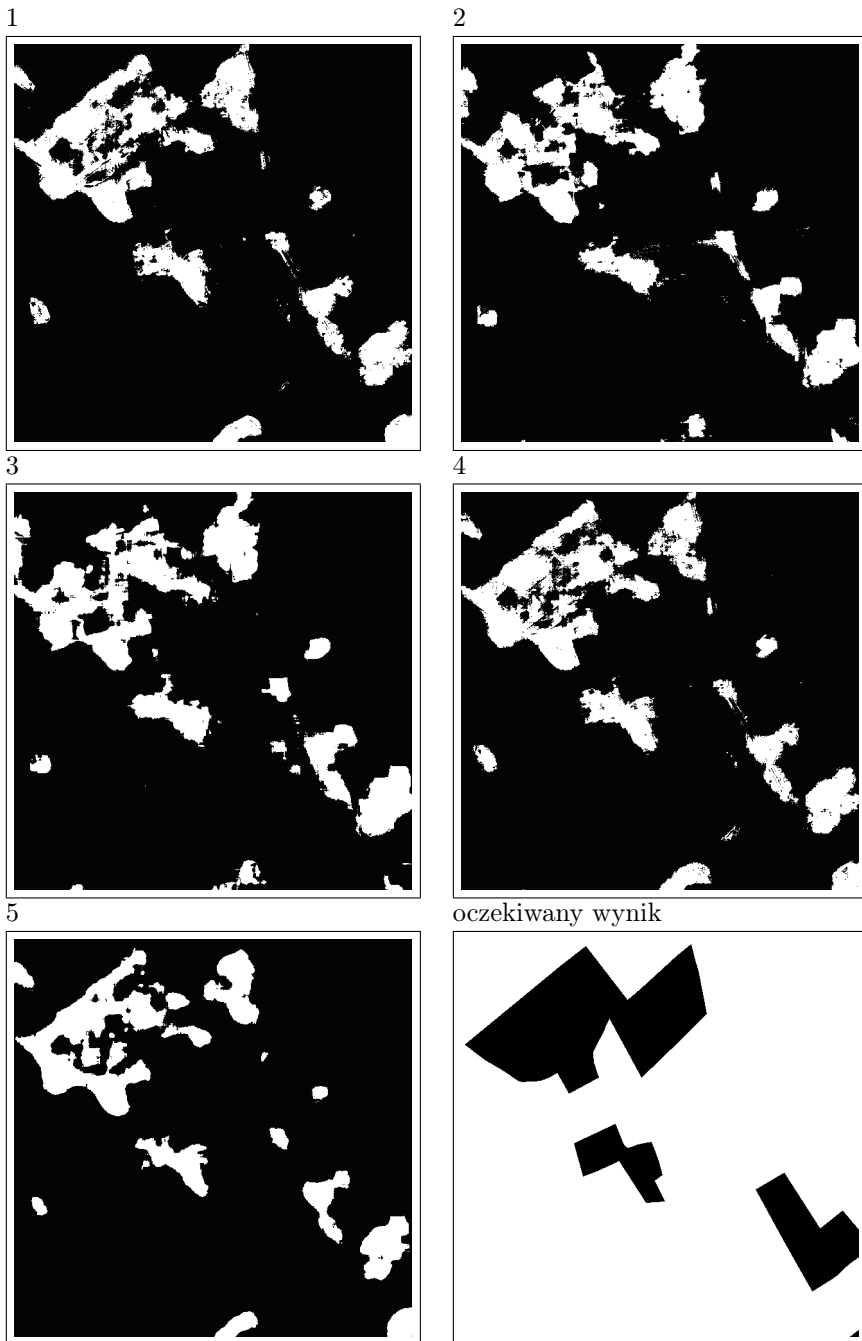
Wyniki klasyfikacji dla obrazka weryfikującego wg powyższych oznaczeń ilustruje rysunek 4.14.

Dla otrzymanych rezultatów przeprowadzono ocenę jakości klasyfikacji. Wyniki przedstawiono w poniższej tabeli. Wartości w tabeli stanowią miary oceny wyników klasyfikacji dla zbioru uczącego i testowego. Indeksy w macierzy niezgodności A : i — klasa otrzymana, j — klasa dostarczona.

Ocena wyników klasyfikacji								
Zest.	Eksp.	κ_w	u_1	u_2	p_1	p_2	d	uDW
Uczący	1	0.6717	0.7832	0.9772	0.6133	0.9899	0.9687	0.0399
	2	0.7789	0.8571	0.9842	0.7333	0.9927	0.9781	0.0264
	3	0.7859	0.8004	0.9881	0.7953	0.9882	0.9773	0.0353
	4	0.6759	0.7802	0.9778	0.6219	0.9895	0.9688	0.0403
	5	0.6339	0.7471	0.9752	0.5782	0.9883	0.9652	0.0463
Kontrolny	1	0.5944	0.6922	0.9633	0.5670	0.9783	0.9457	0.0574
	2	0.5663	0.6418	0.9626	0.5603	0.9731	0.9457	0.0659
	3	0.5874	0.6383	0.9659	0.6010	0.9707	0.9414	0.0660
	4	0.5966	0.6815	0.9642	0.5788	0.9767	0.9452	0.0590
	5	0.5684	0.7050	0.9595	0.5182	0.9814	0.9447	0.0559



Rys. 4.13. Klasyfikacja obrazu uczącego oraz jej oczekiwany wynik.



Rys. 4.14. Klasyfikacja obrazu testowego oraz jej oczekiwany wynik.

4.3.2.1. Wnioski

Niejednorodność w cechach spektralnych, zróżnicowanie po względem częstotliwości jasności pikseli sprawia, że klasa *obszary zabudowane* jest trudna do automatycznej identyfikacji na zdjęciach lotniczych. Przeprowadzony eksperyment miał pozwolić odpowiedzieć na pytanie, na ile reprezentatywne dla tej klasy są cechy GLCM opisujące teksturę i czy w połączeniu z odbiciem światła pozwolą na zbudowanie modelu sprawnego klasyfikatora. Otrzymane wyniki potwierdzają to, iż ekstrakcja cech tekstur jest niezwykle pomocna i bez niej prawidłowe rozpoznanie jest bardzo trudne. Wachlarz przeprowadzonych testów pozwala wnioskować, że istnieją jeszcze możliwości poprawy jakości rozpoznania tej klasy poprzez dalsze badania nad doбором parametrów klasyfikacji.

4.4. Porównanie klasyfikacji obszarów na zdjęciach lotniczych za pomocą masek oraz cech GLCM

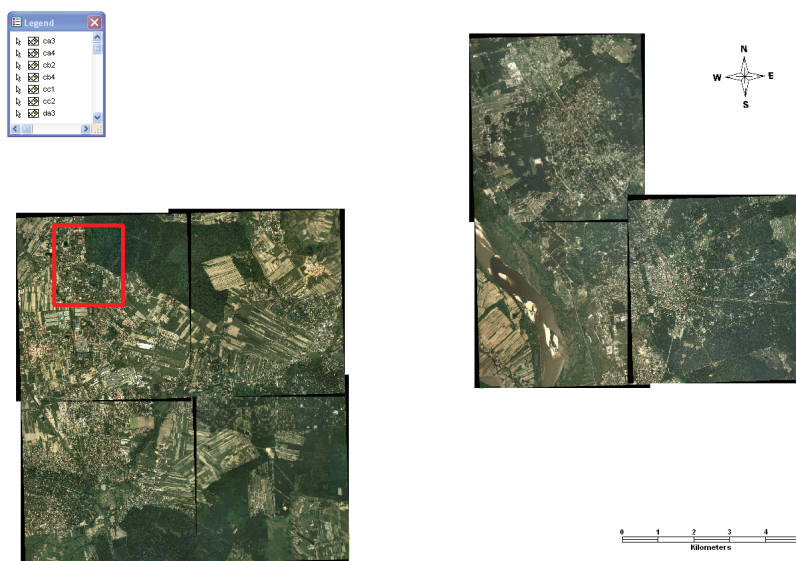
4.4.1. Założenia eksperymentu

Przeprowadzone wcześniej eksperymenty polegające na detekcji/klasyfikacji obiektów powierzchniowych na zdjęciach lotniczych (por. rozdz. 4.2.1) uwzględniały w wektorze cech sąsiedztwo klasyfikowanego piksela, wykorzystując koncepcję maski przemiatającej obraz rastrowy i klasyfikującej piksele za pomocą sieci neuronowej. Takie jednoetapowe podejście zwiększa znacząco złożoność obliczeniową procesu. Dla maski o rozmiarze np. 25x25 pikseli warstwa wejściowa sieci ma rozmiar 1875 neuronów. Sieć, w której istnieją wszystkie możliwe połączenia, a więc powiązanie każdego neuronu warstwy wejściowej ze wszystkimi neuronami warstwy kolejnej itd., wymaga dużych nakładów obliczeniowych na jej nauczanie, a zwiększanie rozmiaru maski powoduje eksponencjalny wzrost złożoności.

W ujęciu dwuetapowym (por. rozdz. 4.3) zastosowano wstępne przetworzenie obrazu wejściowego, polegające na ekstrakcji wektora cech opisujących teksturę w zadanym obszarze. Obszar ten odgrywa podobną rolę co maska w przetwarzaniu jednoetapowym i może również być nazwany maską. Rozmiar maski GLCM nie ma wpływu na złożoność obliczeniową związaną z uczeniem sieci neuronowej, a jedynie na czas trwania obliczeń teksturowych. Wektor cech tekstury składający się z 7 wartości rzeczywistych dla każdego piksela dołączony został w całości do wektora jasności RGB użytego w klasyfikacji z maskami. Otrzymany wektor 9 liczb stanowił wejście sieci neuronowej, przez co rozmiar warstwy wejściowej zredukował się do 9 neuronów. Aby przeprowadzić ocenę jakości tych dwóch podejść, autorzy przeprowadzili kolejny eksperyment. Jako materiał klasyfikacji posłużyły obrazy lotnicze południowych okolic Warszawy o rozdzielczości terenowej około 1 m. Dla tego materiału wykonano w sposób manualny wektoryzację następujących klas pokrycia terenu:

- budynki mieszkalne — *budyn_m*
- budynki przemysłowe — *budyn_p*
- drogi asfaltowe — *drogi_u*
- drogi gruntowe — *drogi_g*
- obszary leśne — *las*
- zadrzewienia — *zadrzew*
- woda stojąca — *woda_stoj*
- grunty porośnięte zbożem — *pola_z*
- grunty porośnięte trawą — *pola_l*
- obszary zabudowy mieszkaniowej — *zab_m*
- obszary zabudowy przemysłowej — *zab_p*
- obszary pozostałe — *inne*

Mozaikę obrazów przedstawia rysunek 4.15. Kwadratem zaznaczono obszar uczący w eksperymencie.

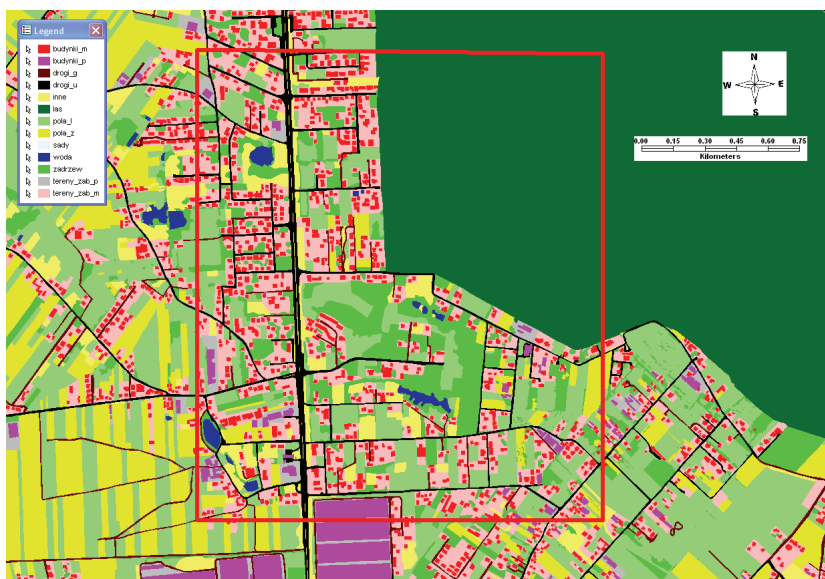


Rys. 4.15. Mozaika obrazów lotniczych użytych w eksperymencie [GUGiK].

Manualną klasyfikację danych eksperymentalnych prezentuje rysunek 4.16, a fragment danych wektorowych zawierających pole uczące — rysunek 4.17.



Rys. 4.16. Manualna klasyfikacja danych eksperymentalnych.



Rys. 4.17. Uczący fragment klasyfikacji danych eksperymentalnych.

4.4.2. Wyniki

Przeprowadzono uczenie sieci dla wybranych klas pokrycia terenu, z trenowaniem oddzielnych sieci dla każdej klasy. W uczeniu tym zastosowano maski o rozmiarach 7×7 , 15×15 oraz 25×25 pikseli. Analizy tekstury dokonano przy następujących ustawieniach parametrów: maska GLCM o rozmiarze 30×30 pikseli i kwantyzacja odcieni szarości do 32 poziomów. Fragmenty ciekawszych wyników klasyfikacji dla zestawu testowego (różnego od uczącego) zaprezentowano na rysunku 4.18.

Obliczone zgodnie z opisaną w rozdziale 1.6 metodą parametry ilościowej oceny jakości klasyfikacji dla wybranych klas zestawiono w poniższej tabeli.

Typ	Sąsiedztwo	κ	u_1	u_2	p_1	p_2	d	uDW
budyn_m	7×7	0.975	0.806	0.977	0.325	0.997	0.975	0.217
	15×15	0.976	0.783	0.979	0.392	0.996	0.976	0.237
	25×25	0.977	0.753	0.982	0.461	0.995	0.977	0.265
	GLCM	0.954	0.694	0.961	0.309	0.992	0.954	0.344
budyn_p	7×7	0.970	0.201	0.983	0.173	0.986	0.970	0.815
	15×15	0.967	0.331	0.992	0.608	0.975	0.968	0.676
	25×25	0.966	0.354	0.992	0.654	0.979	0.967	0.658
	GLCM	0.967	0.524	0.982	0.482	0.984	0.967	0.494
drogi_u	7×7	0.969	0.301	0.972	0.037	0.997	0.969	0.719
	15×15	0.989	0.219	0.990	0.004	0.999	0.990	0.791
	25×25	0.989	0.130	0.990	0.004	1.000	0.989	0.880
	GLCM	0.947	0.244	0.952	0.031	0.995	0.947	0.804
las	7×7	0.857	0.997	0.810	0.635	0.999	0.857	0.192
	15×15	0.926	0.995	0.894	0.815	0.997	0.926	0.111
	25×25	0.939	0.995	0.912	0.849	0.997	0.939	0.093
	GLCM	0.592	0.978	0.342	0.491	0.960	0.592	0.680
pola_l	7×7	0.898	0.788	0.924	0.701	0.949	0.898	0.288
	15×15	0.888	0.759	0.919	0.690	0.941	0.888	0.322
	25×25	0.877	0.717	0.918	0.691	0.926	0.877	0.366
	GLCM	0.867	0.827	0.888	0.791	0.909	0.867	0.285
pola_z	7×7	0.952	0.846	0.954	0.302	0.996	0.952	0.199
	15×15	0.952	0.832	0.974	0.300	0.997	0.952	0.197
	25×25	0.960	0.848	0.971	0.301	0.997	0.960	0.198
	GLCM	0.946	0.816	0.953	0.586	0.984	0.942	0.231
zadrzew	7×7	0.914	0.348	0.941	0.218	0.968	0.915	0.710
	15×15	0.926	0.441	0.937	0.139	0.986	0.926	0.622
	25×25	0.920	0.418	0.946	0.288	0.969	0.920	0.636
	GLCM	0.924	0.757	0.933	0.374	0.986	0.928	0.901
zab_m	7×7	0.793	0.414	0.954	0.793	0.794	0.793	0.629
	15×15	0.861	0.554	0.926	0.615	0.907	0.861	0.519
	25×25	0.891	0.554	0.925	0.615	0.907	0.891	0.520
	GLCM	0.816	0.620	0.896	0.709	0.852	0.816	0.484

Typ	Obraz referencyjny	Wyniki klasyfikacji			
		7×7	15×15	25×25	GLCM
budyn_m					
budyn_p					
drogi_u					
lasy					
pola_l					
pola_z					
zadrzew					
zab_m					

Rys. 4.18. Wyniki klasyfikacji dla zbioru testowego dla różnych rozmiarów masek oraz z użyciem metody GLCM.

4.4.3. Wnioski

Oba testowane podejścia do ekstrakcji cech i klasyfikacji obiektów posiadają podobne możliwości w zakresie jakości otrzymywanych wyników. Zróżnicowanie tematyczne identyfikowanych klas nie pozwala na ekwiwalentne użycie klasyfikacji z maskami i z cechami tekstury. Dla klas obszarowo niewielkich (Budynki, Drogi) podejście GLCM nie poprawia znacząco wyników, natomiast można zauważyć taką poprawę w stosunku do klas o dużej różnorodności w odbiciu spektralnym (obszary zabudowane). Użycie zredukowanego wektora cech dzięki metodzie GLCM znacząco obniża złożoność obliczeniową procesu i przyspiesza wielokrotnie czas uczenia sieci neuronowej. Dla klasy Drogi prezentowane podejścia są nieadekwatne.

Bibliografia

- Abkar A.-A., Sharifi M. A. and Mulder N. J. (2000). Likelihood-based image segmentation and classification: a framework for the integration of expert knowledge in image classification procedures. *International Journal of Applied Earth Observation and Geoinformation* 2(2), pp. 104–119.
- Aplin P. and Atkinson P. M. (2001). Sub-pixel land cover mapping for per-field classification. *International Journal of Remote Sensing* 22(14), pp. 2853–2858.
- Atkinson P. and Lewis P. (2000). Geostatistical classification for remote sensing: an introduction. *Computers and Geosciences* 26(4), pp. 361–371.
- Atkinson P. M. and Emery D. R. (1999). Exploring the relation between spatial structure and wavelength: implications for sampling reflectance in the field. *International Journal of Remote Sensing* 20(13), pp. 2663–2678.
- Baranowski M. and Ciołkosz A. (1997). Opracowanie bazy danych — Pokrycie terenu Polski.
- Barr S. and Barnsley M. (1999). A syntactic pattern recognition paradigm for the derivation of second-order thematic information from remotely sensed images. *in* P. Atkinson and N. Tate (Eds), *Advances in Remote Sensing and GIS Analysis*. Chichester: Wiley. pp. 167–184.
- Bellman C. and Shortis M. (2004). A classification approach to finding buildings in large scale aerial photographs. *in* XXth ISPRS Congress. Istanbul, Turkey.
- Berberoglu S., Lloyd C., Atkinson P. and Curran P. (2000). The integration of spectral and textural information using neural networks for land cover mapping in the Mediterranean. *Computers and Geosciences* 26(4), pp. 385–396.
- Białach M. (2000). Analiza pokrycia terenu Basenu Środkowego Biebrzy na podstawie interpretacji zdjęć satelitarnych wykonanych w latach 1979–1991. *Fotointerpretacja w geografii* 31, pp. 103–114.
- Carr J. R. (1996). Spectral and textural classification of single and multiple band digital images. *Computers and Geosciences* 22(8), pp. 849–865.
- Carr J. R. (1997). Statistical self-affinity, fractal dimension, and geologic interpretation. *Engineering Geology* 48(3-4), pp. 269–282.
- Cheng Q. (1999). Multifractality and spatial statistics. *Computers and Geosciences* 25(9), pp. 949–961.
- Chica-Olmo M. and Abarca-Hernández F. (2000). Computing geostatistical image texture for remotely sensed data classification. *Computers and Geosciences* 26(4), pp. 373–383.

- Clausi D. A. and Zhao Y. (2002). Rapid extraction of image texture by co-occurrence using a hybrid data structure. *Computers and Geosciences* 28(6), pp. 763–774.
- Cohen J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, pp. 37–46.
- Derin H. and Elliott H. (1987). Modelling and segmentation of noisy textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9(1), pp. 39–55.
- Fleiss J. L. (1981). *Statistical Methods for Rates and Proportions*. Vol. 20 of *Educational and Psychological Measurement*. John Wiley & Sons. New York.
- Foody G. M. (2000). Estimation of sub-pixel land cover composition in the presence of untrained classes. *Computers and Geosciences* 26(4), pp. 469–478.
- Geman S. and Geman D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6(6), pp. 721–741.
- German G. W. H. and Gahegan M. N. (1996). Neural network architectures for the classification of temporal image sequences. *Computers and Geosciences* 22(9), pp. 969–979.
- Hammersley J. M. and Clifford P. (1971). Markov fields on finite graphs and lattices. unpublished.
- Haralick R. (1979). Statistical and structural approaches to texture. *Proceedings of IEEE* 67(5), pp. 786–803.
- Haralick R. M. (1986). Texture analysis. *in* T. Y. Young and K.-S. Fu (Eds), *Handbook of pattern recognition and image processing*. Academic Press. San Diego. chapter 11.
- Haralick R. and Shapiro L. (1992). *Computer and Robot Vision*, vol. 1.. Vol. 1. Addison-Wesley Publishing. Reading, MA.
- Hubert-Moy L., Cotonnac A., Du L. L., Chardin A. and Perez P. (2001). A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote Sensing of Environment* 75(2), pp. 174–187.
- Höppner F. (2000). Fuzzy clustering algorithms — a tool library. Technical report. University of Applied Sciences Oldenburg/Ostfriesland/Wilhelmshaven. Braunschweig, Germany.
URL: <http://public.fh-wolfenbuettel.de/~hoeppnef/download/fc-0.3.7-gcc-4.1.2.tar.gz>

- Höppner F., Klawonn F., Kruse R. and Runkler T. (1999). *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley.
- Jakomulska A. (1998). Zastosowanie logiki rozmytej w klasyfikacji nadzorowanej. Fotointerpretacja w geografii .
- Ju J., Kolaczyk E. D. and Gopal S. (2003). Gaussian mixture discriminant analysis and sub-pixel land cover characterization in remote sensing. *Remote Sensing of Environment* 84(4), pp. 550–560.
- Keuchel J., Naumann S., Heiler M. and Siegmund A. (2003). Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data. *Remote Sensing of Environment* 86(4), pp. 530–541.
- Kozubek E. (2000). Zmiany użytkowania ziemi w rejonie tarnobrzeskim pod wpływem uprzemysłowienia w latach 1937–1992 w świetle interpretacji map i obrazów satelitarnych. Rozprawa doktorska. Warszawa.
- Landis J. and Koch G. (1977). The measurement of observer agreement for categorical data. *Biometrics* 33, pp. 159–174.
- Li S. Z. (1995). *Markov Random Fields: Theory and Application*. Academic Press. Boston.
- Lira J. and Maletti G. (2002). A supervised contextual classifier based on a region-growth algorithm. *Computers and Geosciences* 28(8), pp. 951–959.
- Liu X.-H., Skidmore A. and Oosten H. V. (2002). Integration of classification methods for improvement of land-cover map accuracy. *ISPRS Journal of Photogrammetry and Remote Sensing* 56(4), pp. 257–268.
- Meng L. (1993). Application of neural network in cartographic pattern recognition. *in Proceedings 16th International Cartographic Conference*. Vol. 1. Cologne. pp. 192–202.
- Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H. and Teller E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21, pp. 1087–1091.
- Miller D. M., Kaminsky E. J. and Rana S. (1995). Neural network classification of remote-sensing data. *Computers and Geosciences* 21(3), pp. 377–386.
- Müller J.-C. (1992). Parallel distributed processing: An application to geographic feature selection. *in Proceedings Fifth International Symposium on Spatial Data Handling*. Vol. 1. Charleston, SC. pp. 230–240.
- Openshaw S., Wymer C. and Cross A. (1991). Using neural nets to solve some hard analysis problems in GIS. *in Proceedings EGIS '91*. Vol. 2. Bruxelles. pp. 797–807.

- Paget R. D. (1999). Nonparametric Markov Random Field Models for Natural Texture Images. Ph.d. thesis. Cooperative Research Centre for Sensor Signal and Information Processing, Department of Computer Science & Electrical Engineering, The University of Queensland. St Lucia, Queensland 4072, Australia.
- Pal M. and Mather P. M. (2003). An assessment of the effectiveness of decision tree methods for land cover classification. *Remote Sensing of Environment* 86(4), pp. 554–565.
- Ruiz-del Solar J. (1998). TEXSOM: Texture segmentation using self-organizing maps. *Neurocomputing* 21, pp. 7–18.
- Ryherd S. and Woodcock C. (1996). Combining spectral and texture data in the segmentation of remotely sensed images. *Photogramm. Eng. Remote Sens.* 62, pp. 181–194.
- Stuckens J., Coppin P. and Bauer M. (2000). Integrating contextual information with per-pixel classification for improved land cover classification. *Remote Sensing of Environment* 71(3), pp. 282–296.
- Tatem A., Lewis H., Atkinson P. and Nixon M. (2001). Super-resolution target identification from remotely sensed images using a hopfield neural network. *IEEE Transactions on Geoscience and Remote Sensing* (39), pp. 781–796.
- Tuceryan M. and Anil K. (1998). Texture analysis. *in* C. Chen, L. Pau and P. Wang (Eds), *The handbook of Pattern recognition and computer vision*. 2nd edn. World Scientific Publishing Co.. chapter 2.1, pp. 207–248.
- Venkatesh Y. and Raja S. K. (2003). On the classification of multispectral satellite images using the multilayer perceptron. *Pattern Recognition* 36(9), pp. 2161–2175.
- Yu Y. and Cheng Q. (2003). Rf parameter estimation by an accelerated method. *Pattern Recognition Letters* 24, pp. 1251–1259.
- Zhang Y. (1999). Optimisation of building detection in satellite images by combining multispectral classification and texture filtering. *ISPRS Journal of Photogrammetry and Remote Sensing* 54(1), pp. 50–60.

RASTER IMAGE CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS AND STATISTICAL CLASSIFICATION METHODS

S u m m a r y

This publication is the result of several years of research* on processing raster images — both aerial photographs and satellite images — using a broad spectrum of image processing methods and statistical classification methods.

In particular, in many cases of classification, extensive use has been made of artificial neural networks, whose principle advantages are:

- the ability to process data in the situations, where there is no correct formal model, or, generally, a good understanding of the problem is lacking, as well as any kind of heuristic information,
- the ability of learning algorithms of data processing without programming, based on a series of examples,
- the ability to generalize,
- resistance to errors and other glitches in data (such as some data missing, etc.).

During the last decade the geographic information systems in Poland have undergone significant changes, namely, aerial and satellite images have begun to be used routinely. One of the factors motivating these changes was joining the European Union and the implementation of the IACS system, which requires covering the whole country by high-resolution aerial — and partially satellite — images. In addition to being used for the purposes of verifying the direct subsidies for the agriculture, these images have found a number of other uses. One of them was building the national spatial data infrastructure (NSDI): the topographic database (TDB) with the level of detail corresponding to the 1:10,000 scale, and the vector map level two (VMAP L2) with the level of detail corresponding to the 1:50,000 scale.

The original premise of the work undertaken in the project was to develop the technology of applying the artificial neural networks to provide a data feed and verification ability for the referential data bases using the aerial photographs and satellite images. The initial series of experiments used image processing with the aid of a mask (the immediate neighborhood of a pixel, explained in more detail in chapters 2 and 4) which was used as input data to the neural network (see

*This work has been supported from the funds of the Committee of Scientific Research under the research project number 5T12E02924 during the years 2003–2006.

sections 1.2 and 1.3). But since not all the results obtained from the automatically trained neural networks were satisfactory, a number of fundamental statistical techniques have been applied for comparison, such as cluster analysis (section 1.4), or the Bayesian classifier, so-called “naive” (section 1.5).

During the course of this research, there soon appeared a need to use an objective method for evaluating the quality of the results obtained during the subsequent experiments. Most desirable — from the point of view of the area classification — would be to have an objective and global measure of the quality of classification using the given method. For this purpose an analysis was performed of the better known quality measures of this type, and section 1.6 contains their review and discussion. It turned out that no single evaluation method provides an absolute and adequate measure of the classification quality. Therefore in the following part of this monograph a complete set of them is calculated and presented for all the experimental results.

The subsequent work undertook a classical task of satellite image classification. A number of experiments have been conducted using data from the NOAA, Landsat, and SPOT satellites. For this purpose the image data have been augmented with additional statistical characteristics, allowing to take into account — in addition to the spectral response — the texture of the area under analysis. A number of statistical feature extraction methods have been applied, such as: chromatic moments, Markov and Gibbs random fields, variograms, and the GLCM matrix elements. The theoretical background of these methods have been presented in chapters 2 and 3. The results obtained in this research were mixed. For example, image segmentation using the chromatic moments brought partial success, while the attempt to use the Markov random fields has failed. Some classification results using textures have been very successful, with relative error in the neighborhood of a few percent.

The next stage of research concerned processing aerial photographs (chapter 4). This was due to the efforts for developing a method to feed referential databases from aerial photographs. The goal for this work was not to automate the data acquisition process, but just to detect specific objects, which can support the verification of the database creation tasks performed by outside cartographic entities. Additionally, effort was made in this project to determine the built-up areas for the TDB and VMAP Level 2. Since the results obtained using neural networks on just the spectral response have been inadequate, additional processing and advanced spatial analysis was necessary to improve them. In this case the earlier developed methods based on textures, and more specifically, the GLCM method turned out to be particularly useful. This improved the classification results, although the correctness of this classification varied with the objects being classified, from excellent results for classes such as water and forests, to more modest results for buildings and roads.

In most cases the inclusion of the GLCM parameters gave the same or better results as with masks, but at the lower computational costs.

All work in this project has been done using strictly non-commercial software. This includes both complete, ready-to-use applications, such as the excellent neural network simulator SNNS, as well as raster image processing libraries OpenCV and Image Magick, and basic programming tools such as: C and C++ compilers, editors and debuggers, which have been used to develop specific programs for the purpose of this project.