



Uniwersytet Jagielloński w Krakowie

Wydział Fizyki, Astronomii i Informatyki Stosowanej

Krzysztof Kozubek

Nr albumu: 1120405

Rozpoznawanie obrazów z wykorzystaniem biblioteki OpenCV

Praca licencjacka
na kierunku Informatyka

Praca wykonana pod kierunkiem
doktora Wiesława Chmielnickiego
Zakład Technologii Gier

Kraków 2016

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

.....
Kraków, dnia

.....
Podpis autora pracy

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

.....
Kraków, dnia

.....
Podpis kierującego pracą

Spis Treści

WSTĘP	5
1. METODY ROZPOZNAWANIA OBRAZÓW	6
1.1 AKWIZYCJA	6
1.2 PRZETWARZANIE WSTĘPNE	8
1.2.1 Przekształcenia punktowe	8
1.2.2 Przekształcenia geometryczne	9
1.2.3 Przekształcenia kontekstowe	9
1.2.4 Przekształcenia morfologiczne	10
1.3 SEGMENTACJA	10
1.3.1 Segmentacja przez podział obszaru	11
1.3.2 Segmentacja przez rozrost obszaru	12
1.4 ROZPOZNAWANIE I INTERPRETACJA	12
1.4.1 Reprezentacja cechy	12
1.4.2 Selekcja cech	13
1.4.3 Skalowanie cech	14
1.4.4 Kalibracja klasyfikatora	14
1.4.5 Klasyfikacja	16
1.5 WERYFIKACJA	16
2. KLASYFIKATORY	18
2.1 ROZPOZNAWANIE SYNTAKTYCZNE	18
2.1.1 Metody ciągowe	19
2.1.2 Metody drzewowe	20
2.1.3 Metody grafowe	20
2.2 ROZPOZNAWANIE DECYZYJNO-TEORETYCZNE	21
2.2.1 Klasyfikacja nienadzorowana	21
2.2.1.1 Algorytm k-średnich	22
2.2.1.2 Algorytm ISODATA	22
2.2.2 Klasyfikacja nadzorowana	23
2.2.2.1 Sieci neuronowe	23
2.2.2.2 Maszyna wektorów nośnych	25
2.2.2.3 Klasyfikatory minimalno-odległościowe	26
2.2.2.4 Drzewa decyzyjne	27
2.2.2.5 Naiwny klasyfikator Bayesa	28

3.	BIBLIOTEKA OPENCV	30
3.1	INSTALACJA BIBLIOTEKI OPENCV	30
3.1.1	<i>Instalacji biblioteki na Linux</i>	<i>30</i>
3.1.2	<i>Instalacja na systemie Windows</i>	<i>30</i>
3.2	MODUŁ PODSTAWOWY	30
3.2.1	<i>Reprezentacja zdjęcia</i>	<i>30</i>
3.2.2	<i>Operacje na zdjęciu</i>	<i>31</i>
3.3	PRZETWARZANIE OBRAZU	31
3.3.1	<i>Operacje morfologiczne</i>	<i>32</i>
3.3.2	<i>Krawędzie</i>	<i>34</i>
3.3.3	<i>Narożniki</i>	<i>35</i>
3.3.4	<i>Przekształcenia</i>	<i>35</i>
3.4	INTERFEJS UŻYTKOWNIKA.....	36
3.5	HISTOGRAM.....	36
3.6	ZAAWANSOWANE FUNKCJE OPENCV.....	37
3.6.1	<i>Podstawowe operacje na filmach</i>	<i>37</i>
3.6.2	<i>Ekstrakcja</i>	<i>37</i>
3.6.3	<i>Rozpoznawanie twarzy</i>	<i>37</i>
4.	LIBSVM.....	38
5.	PRZEPROWADZENIE EKSPERYMENTU.....	40
5.1	PRZYGOTOWANIE ZBIORU DANYCH.....	40
5.2	SEGMENTACJA.....	42
5.3	NAUKA KLASYFIKATORA	44
5.4	WYNIKI EKSPERYMENTU.....	46
6.	PODSUMOWANIE	47
7.	BIBLIOGRAFIA	49

Wstęp

Metody rozpoznawania obrazów są jedną z najszybciej rozwijających się dziedzin, w których wykorzystywane są komputery. Powodem jest wzrastające zapotrzebowanie na analizę ogromnej i ciągle wzrastającej ilości danych w formie filmów, zdjęć, obrazów itp. Analizy tej ze względu na ilość danych nie jest już w stanie dokonać człowiek. Stąd pomysł na wykorzystanie maszyn, które ze względu na swoją szybkość działania i możliwość ciągłej pracy idealnie nadają się do tego zadania. Nie bez znaczenia jest także to, iż maszyny te mogą pracować tam, gdzie człowiek nie może: na przykład w środowisku niebezpiecznym lub niedostępnym. Wreszcie koszty utrzymania są często aspektem decydującym o wyborze maszyn do tej roli.

Metody te są szczególnym przypadkiem metod rozpoznawania wzorca. Skupiają się na analizie danych znajdujących się na obrazach statycznych jak i dynamicznych. Przewagą metod rozpoznawania obrazów nad dopasowaniem wzorca jest możliwość określenia podobieństwa i przynależności obiektu do danej klasy jako wartości prawdopodobieństwa. Powoduje to stosowanie metod rozpoznawania obrazów wszędzie gdzie obiekty różnią się od siebie lub jest małe prawdopodobieństwo na znalezienie identycznych obiektów.

W dużym uproszczeniu, dwoma głównymi zadaniami rozpoznawania obrazów są akwizycja obrazu oraz jego rozpoznanie. Akwizycja ma za zadanie pobranie informacji dotyczącej badanego obiektu. Wynikiem tego procesu jest otrzymanie zdjęcia lub video w wersji cyfrowej. Po przekazaniu informacji o obiekcie do pamięci komputera następuje proces rozpoznawania. Proces ten polega na obróbce obrazu, składającego się z wstępnego przetwarzania oraz segmentacji. Efektem tych działań jest otrzymanie reprezentacji obiektów znajdujących się na zdjęciu lub filmie. Z danych obiektów zostają wyselekcjonowane cechy, które zostają analizowane oraz dzięki którym możliwe jest rozpoznanie danego obiektu.

Przykładami wykorzystania metod rozpoznawania obrazów znanym każdemu z nas mogą być np. analiza zdjęć samochodowych wykonanych przez fotoradary, wykrywanie twarzy osób poszukiwanych w nagraniach monitoringu, czy rozpoznawanie tekstu drukowanego lub odręcznego ze zdjęcia zrobionego telefonem. Ostatni przykład pokazuje, że metody rozpoznawania obrazów, nie są już wykorzystywane tylko przez przemysł czy służby porządkowe – są (świadomie lub nie) wykorzystywane przez każdego z nas.

1. Metody rozpoznawania obrazów

Zautomatyzowane systemy rozpoznawania obrazów mają wielki wpływ na życie ludzkie. Wykorzystywane są w celu dokonywania wyboru lub pomocy w podjęciu decyzji, poprzez analizy obrazów takich jak: RTG, USG, Tomografii itp.

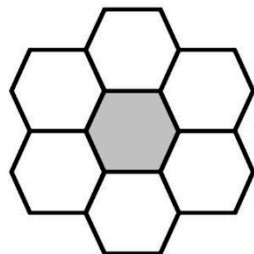
Metody rozpoznawania obrazów dzielą się na trzy elementy składowe:

- Przetwarzanie niskiego poziomu: obejmuje akwizycję, przetwarzanie wstępne, poprawę jakości obrazu.
- Przetwarzanie średniego poziomu: obejmuje segmentację, wydzielenie i opis obiektów znajdujących się na obrazie
- Przetwarzanie wysokiego poziomu: obejmuje klasyfikację i interpretację analizowanych scen.

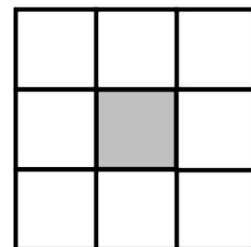
1.1 Akwizycja

Pierwszym etapem rozpoznawania obrazu jest akwizycja obrazu, oznaczająca ucyfrowienie rzeczywistej sceny. Etap ten ma bardzo duży wpływ na kolejne kroki rozpoznawania obrazu, dlatego ważne jest by dobrać odpowiedni sprzęt i dobrze przygotować go do akwizycji.

Obecnie do ucyfrowienia rzeczywistej sceny wykorzystuje się dwa sposoby rozmieszczenia cyfrowych elementów obrazu (pikseli): według siatki kwadratowej lub heksagonalnej. Siatka heksagonalna jest zbliżona do układu receptorów wzroku człowieka, zaś siatka kwadratowa jest dużo prostsza i wygodniejsza, stąd też częściej jest wykorzystywana i przypomina siatkę składającą się z wielu kwadratów.



Rysunek 1. Siatka heksagonalna



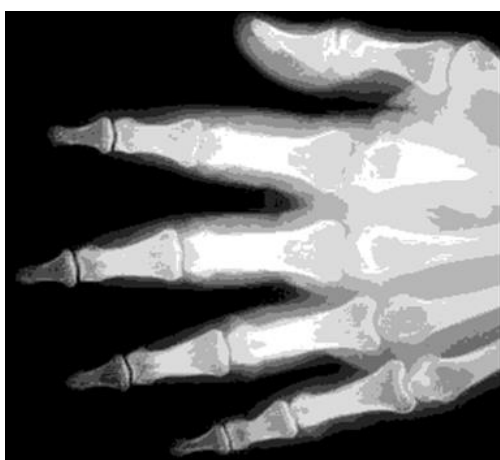
Rysunek 2. Siatka kwadratowa

Najważniejszymi parametrami reprezentacji elementów obrazu są:

- rozdzielczość obrazu - wpływa na ilość zawartych informacji, jakość obrazu oraz szybkości przetwarzania obrazu. Wartość podawana jest w liczbie pikseli w poziomie i pionie np. 640 x 480, 800 x 600, 1280 x 720, 1920 x 1080.
- ilość kolorów - wpływa na jakość obrazu i zajmowaną pamięć. Np. binarny (1 bpp¹), monochromatyczny (8 bpp), kolorowy (24 lub 32⁸) i wiele innych pośrednich formatów.

Przykłady doboru parametrów:

- Głębia kolorów

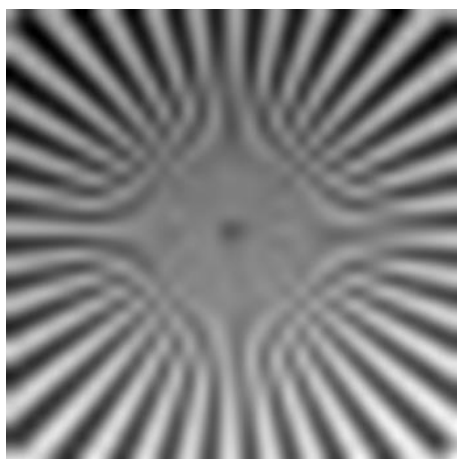


Rysunek 3. Zdjęcie z 8 odcieniami szarości

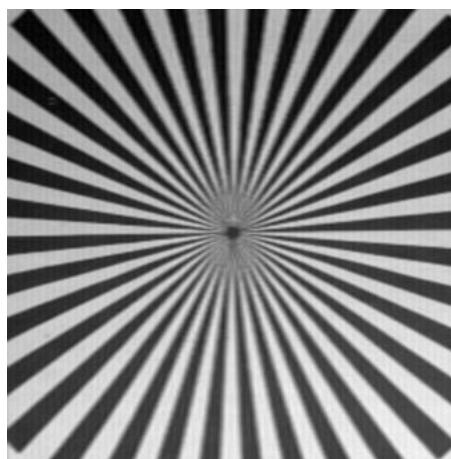


Rysunek 4. Zdjęcie z 256 odcieniami szarości

- Rozdzielczość



Rysunek 5. Rozdzielczość 10 dpi



Rysunek 6. Rozdzielczość 72 dpi

¹ BPP (ang. *Bits Per Pixel*) – liczba bitów przeznaczona do reprezentacji koloru piksela.

Przykłady urządzeń służących do akwizycji obrazu:

- Kamera video CCD (ang. *Charge Coupled Device*) – zaletą jest dynamiczny obraz, duża prędkość przetwarzania, mały koszt, gabaryty przyjazne do przemieszczania, wadą jest mała rozdzielczość.
- Aparat fotograficzny – zalety to: duża rozdzielczość i małe gabaryty.
- Skaner – zaletą jest wysoka rozdzielczość, wadami natomiast są długi czas przetwarzania, cena i gabaryty.
- Tworzenie obrazu przy pomocy programu – zaletami jest cena i kontrola tworzenia obrazu.

Ucyfrowione sceny mogą być analizowane na dwa sposoby: dynamicznie lub statycznie. Analizowanie obrazu dynamicznego jest w rzeczywistości analizą obrazu video, która pozwala na korelowanie danych obecnych z wcześniejszymi.

1.2 Przetwarzanie wstępne

Przetwarzaniem wstępnym nazywamy każdą operację dążącą do poprawienia jakości obrazu lub operację której celem jest uwydatnienie obiektu. W celu zmniejszenia zniekształceń powstałych wskutek akwizycji obrazu, korzystamy z różnych rodzajów przetwarzania wstępnego.

1.2.1 Przekształcenia punktowe

Przekształcenie punktowe to przekształcenia elementów (pikseli) obrazu, poprzez dokonanie na nich pewnych operacji niezależnie od stanu elementów sąsiadujących. Dzięki temu operacje są bardzo proste i szybkie. Najprostszymi operacjami są: tworzenie negatywu, przyciemnianie lub rozjaśnianie obrazu.

- Binarizacja to prawdopodobnie najważniejsza metoda przetwarzania punktowego obrazu, występująca niemal zawsze. Celem jest redukcja części informacji zawierających się na obrazie. Proces polega na zmianie odcieni szarości do 2 (0 lub 1). Binarizację przeprowadzać można na wiele sposobów, w bibliotece OpenCV funkcje tą można uzyskać wykorzystując metody: *threshold*, *adaptiveThreshold* z metodą *bitwise_and*.
- Wyrównaniem histogramu nazywamy przekształcenie jasności punktów obrazu w taki sposób, aby histogram był jak najbardziej płaski. Wynikiem jest uwypuklenie danych elementów na obrazie, które są słabo widoczne przez mały kontrast. Biblioteka OpenCV posiada funkcję *equalizeHist* odpowiedzialną za wyrównanie histogramów.

1.2.2 Przekształcenia geometryczne

Do przekształceń geometrycznych zalicza się: odbicia, obroty, przesunięcia i inne. Celem przekształceń geometrycznych jest korekcja błędów takich jak: zniekształcenia poduszkowe, beczkowate czy trapezowe. Biblioteka OpenCV posiada zaimplementowaną metodę *warpAffine* do przesuwania, obracania i odbijania obrazu.

1.2.3 Przekształcenia kontekstowe

Przekształcenia kontekstowe wykorzystujące filtry, polegają na modyfikacji elementów (pikseli) obrazu będących w zależności od stanu ich samych oraz ich otoczenia. Operacje te mogą wykonywać się przez długi czas ze względu na kontekstowość.

Aby otrzymać wynik dla jednego punktu należy poddać operacji ten punkt jak również kilka punktów go otaczających. Algorytmy filtracji są proste dlatego możliwe jest by przekształcenie wykonywało się jednocześnie w kilku miejscach na raz.

Przekształcenie nie może jednak wykonywać się na brzegach zdjęcia z powodu braku informacji o stanie sąsiadujących punktów. Do podstawowych przekształceń kontekstowych zaliczamy:

- Filtry liniowe są liniowe wtedy i tylko wtedy gdy spełniają dwie podstawowe reguły:

Funkcja jest addytywna: $\varphi(f + g) = \varphi(f) + \varphi(g)$, oraz

funkcja jest jednorodna: $\varphi(\lambda f) = \lambda \varphi(f), \lambda \in \mathbb{R}$

Gdzie: φ – funkcja realizująca filtr,

f, g – obrazy poddawane filtracji.

Filtry liniowe wykorzystuje się ze względu na łatwą implementację oraz ze względu na to, że obrazy wynikowe są często intuicyjną wartością wynikową. W celu skorzystania z filtrów w bibliotece OpenCV, należy użyć metody *filter2D*.

- Filtry dolnoprzepustowe, najczęściej stosujemy do odsumowania obrazu.

Przykład: prosty filtr uśredniający, którego macierz konwolucji ma postać:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Zastosowany filtr rozmywa obraz, czego skutkiem jest zniknięcie drobnych zakłóceń. Filtr ten również zniekształca obiekty przez rozmycie konturów.

- Filtry górnoprzepustowe (inaczej nazywane gradientami), głównym celem jest wydobycie z obrazu takich elementów jak krawędzie. Bardzo popularne jest mówienie o filtrach górnoprzepustowych jako dokonujących wyostrenia sygnału, czyli uwypuklenia krawędzi obiektów. Typowymi maskami stosowanymi są:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 7 & -1 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 1 & -9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

1.2.4 Przekształcenia morfologiczne

Do najważniejszych operacji służących do analizy komputerowej zaliczane są przekształcenia morfologiczne. Wynik tych przekształceń powoduje zmianę struktury lub formy obiektu. Operacje te posiadają wielką złożoność obliczeniową z powodu wykonywania operacji typu: analiza kształtu, wzajemnego położenia obiektów itp.

Operacje morfologiczne opierają się na elementach strukturalnych obrazu, będących pewnym fragmentem obrazu. Najczęściej przyjmują one postać kwadratu, prostokąta lub koła można jednak spotkać inne nieregularne kształty.

- Erozja nadaje wartość minimalną intensywności dla punktu będącego w otoczeniu określonej macierzy elementu. W obrazach binarnych punkt (piksel), przyjmuje wartość 0 gdy stan choć jednego z punktów w jego otoczeniu posiada wartość 0.
- Dylatacja jest odwrotnością erozji. Nadaje wartość maksymalną intensywności dla punktu będącego w otoczeniu określonej macierzy elementów. W obrazach binarnych punkt (piksel), przyjmuje wartość 1 gdy stan choć jednego z punktów w jego otoczeniu posiada wartość 1.
- Otwarcie i zamknięcie opierają się na wyżej wspomnianych przekształceniach erozji i dylatacji. Celem powstania tych przekształceń była zmiana pola powierzchni przekształceń obrazu. Otwarciem nazywamy procesy najpierw erozji a potem dylatacji natomiast zamknięciem jego odwrotność czyli najpierw proces dylatacji a potem erozji.

1.3 Segmentacja

Segmentacją obrazu jest podział obrazu na spójne fragmenty, różniące się o siebie. Często towarzyszy temu procesowi indeksacja obiektów, polegająca na przypisaniu każdemu z pikseli odpowiedniej etykiety, która określa do jakiego obiektu może on należeć. Segmentacja to nic innego jak wydzielenie obszarów obrazów spełniających

pewne kryteria jednorodności. Przykładem takich kryteriów mogą być: kolor obrazu lub poziom jasności. Można wyróżnić dwie techniki segmentacji [31].

1.3.1 Segmentacja przez podział obszaru

Metoda ta polega na porównywaniu wartości każdego elementu obrazu z wartością progową (T). Porównywany punkt zostaje przydzielony do jednej z dwóch grup: przekraczającej (punkty obiektu) lub nieprzekraczającej (punkty tła) wartości progowej. Wartość progową dobieramy zwykle na podstawie histogramu. Po przeprowadzeniu procesu segmentacji otrzymuje się tzw. obraz binarny.

Zakładając, że g jest obrazem wynikowym, $f(x,y)$ zwraca wartość piksela obrazu wejściowego w punkcie (x,y) oraz oznaczając wartość progu przez literę T , to możemy zdefiniować progowanie:

$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) \geq T \end{cases}$$

Podczas korzystania z tej metody segmentacji mogą wystąpić trudności z doбором odpowiedniego progu dyskryminacji. Zwłaszcza gdy krzywa histogramowa jest wielobarbowa, a należy przeprowadzić separacje między wierzchołkami. Powstaje problem doboru odpowiedniego progu.

Drugą trudnością są pojawiające się tzw. fałszywe obiekty. Rozwiązaniem tego jest zastosowanie tzw. „określenia obszaru jednolitego”. Polega to na dodawaniu pikseli, które spełniają warunki progowania oraz będą sąsiadami jednego lub więcej pikseli należących już do obszaru. Metoda znajduje zastosowanie dla obiektów jednorodnych z kontrastowym tłem, posiada jednak zastosowanie również dla złożonych scen.

Segmentacja z stałym progiem jasności dla całego obrazu nie jest skuteczna i jest nazywana segmentacją wykorzystującą próg globalny. Częściej wykorzystywana jest segmentacja z progiem zależnym od wartości przestrzennych (x,y) , taki próg jasności nazywamy dynamicznym. Trzecim typem progu jasności jaki jest stosowany to próg lokalny, jego wartość jest wyliczana wzorem:

$$T = T[p(x,y), f(x,y)]$$

gdzie $p(x,y)$ to cechy obrazu (np. średnia jasność obrazu w pewnym otoczeniu).

W przypadku segmentacji progowania (wielopoziomowego) obrazu z większą ilością jasności ($N + 1$), progowanie przyjmuje postać:

$$g(x, y) = \begin{cases} R1, & f(x, y) \leq T_1 \\ R2, & T_1 < f(x, y) \leq T_2 \\ RN + 1, & T_N < f(x, y) \leq T_{N+1} \end{cases}$$

Najczęściej ogranicza się do trzech kolorów. Spowodowane jest to reprezentacją trójwymiarową przestrzeni kolorów RGB (ang. Red Green Blue).

1.3.2 Segmentacja przez rozrost obszaru

Segmentacja przez rozrost obszaru polega na znalezieniu wszystkich grup elementów, które posiadają podobne kryteria jednorodności.

Najprostszą metodą tej segmentacji jest przyjęcie punktu początkowego zwanego ziarnem i sprawdzanie elementów sąsiadujących czy posiadają zbliżoną jasność. Elementy sąsiadujące, które spełniają to kryterium jednorodności zostają przypisane do grupy do której należy obiekt z którym porównujemy element.

Proces jest powtarzany aż wszystkie elementy zostaną przypisane do pewnych grup. Wynikiem tego procesu jest podzielenie obrazu na grupy.

Najczęściej za kryteria jednorodności, przyjmuje się miary podobieństwa wektora cech danego obszaru. Przykładowymi miarami podobieństwa:

- Iloczyn skalarny wektorów: $\vec{x}_i \circ \vec{x}_j = |\vec{x}_i| * |\vec{x}_j| * \cos \alpha$
- Odległość Euklidesową: Tabela 1.

1.4 Rozpoznawanie i interpretacja

Ostatnim etapem rozpoznawania obrazu, który zwraca wynik całego procesu jest klasyfikacja i interpretacja obiektów. Istnieje wiele algorytmów klasyfikacji, część z nich opiera swoje działanie na wyliczonych wartościach cech z obiektu. Cechy te są wyliczane z cech kształtu obiektu, które możemy podzielić na [38]:

- Topologiczne (spójność, liczba Eulera, liczba otworów, szkielet),
- Geometryczne (obwód, pole powierzchni, promień max-min, kołowość, symetria, środek ciężkości, rzuty Fereta, prostokąt opisany, dopasowana elipsa).

1.4.1 Reprezentacja cechy

Cechy mogą być reprezentowane za pomocą [26]:

- Wartości mierzalnych (liczb)
 - Skokowa / dyskretna (zbiór wartości przeliczalnych np. liczba kwiatów)
 - Ciągła (liczby rzeczywiste np. czas oczekiwania)
- Niemierzalnych (wyrażona za pomocą słów)
 - Porządkowa (możliwe ustalenie relacji porządku pomiędzy elementami np. wykształcenie)
 - Nominalna (brak porządku np. kolor liści)

Cechy najczęściej są reprezentowane przez wartości mierzalne. Wektor m wymiarowych cech przedstawiający obiekt reprezentowany przez obraz nazywamy wzorcem:

$$\vec{x}_k = [x_1, x_2, \dots, x_m],$$

Gdzie: k - numer wzorca, x - cecha / atrybut, m - numer składowej.

Ważne jest by cechy wyróżniały obiekty z różnych klas, były jak najbardziej podobnymi wartościami dla tych samych klas a także były odporne na zakłócenia, zmianę położenia, skalowania czy orientację obiektu. Mając na uwadze szybkość działania klasyfikacji, ważne jest by eliminować cechy zawierające informacje redundantne lub informacje nieistotne, które zwalniałyby ten proces.

1.4.2 Selekcja cech

Selekcja cech polega na wyborze cech (atrybutów obiektu), zawierających pewne informacje o obiekcie pod kątem przydatności podczas procesu klasyfikacji. Ma to na celu zwiększyć jakość klasyfikacji oraz zmniejszyć czas jej działania.

Algorytmy selekcji cech podawanych [16][17][35] można podzielić na:

- Metody rankingowe polegają na przypisaniu każdej z cech współczynnika przydatności [18][36], nie uwzględniając zależności pomiędzy nimi. Do tego nie jest wykorzystywany klasyfikator, dlatego rozwiązanie to jest uniwersalne dla każdego klasyfikatora. Po dokonaniu oceny cech, wybór cech może przyjąć dwie formy: poprzez wybranie stałego progu mówiącego o liczbie wyboru najlepszych cech lub dobierać po kolei najlepsze cechy do czasu aż wynik klasyfikacji będzie się poprawiać.
- Metody opakowane wybierają podzbiór cech, który zostaje wykorzystany przez klasyfikator [19]. W oparciu o wynik podzbiór jest modyfikowany i oceniany przez klasyfikator. Proces jest powtarzany do wyznaczenia podzbioru z cechami końcowymi. Wybór cech jest uzależniony od wybranego klasyfikatora. Metody te wielokrotnie wykorzystują klasyfikator, co wiąże się z dużym kosztem obliczeniowym.

- Metody hybrydowe łączą metody rankingowe i opakowane. Proces polega na stworzeniu podzbiorów opartych na przydatności cech (redukcja zbioru cech), po czym z wyznaczonych podzbiorów, klasyfikator ocenia cechy [20]. Metody te lepiej radzą sobie z dużymi zbiorami danych.
- Metody wbudowane są wbudowane ściśle w klasyfikator. Metody te są używane w procesie uczenia, podczas tworzenia modelu. Wykorzystywane są do zredukowania wielowymiarowości problemu [18]. Proces selekcji cech nie może zostać pominięty podczas tworzenia modelu [21].

1.4.3 Skalowanie cech

W celu zmniejszenia rzędu wielkości wartości parametrów oraz zapobieganiu zdominowania cech o mniejszych wartościach przez cechy o większe wartości, stosuje się skalowanie wartości. Może ono przyjąć postać:

- Normalizacji, gdzie wszystkie wartości przyjmują wartość z przedziału $[0;1]$. Transformacja jest przeprowadzana za pomocą wzoru:

$$a'_i = \frac{a_i - a_{i_min}}{a_{i_max} - a_{i_min}}$$

Gdzie: a_{i_max} jest maksymalną wartością dla i-tych cechy zaś
 a_{i_min} jest minimalną wartością dla i-tych cech.

- Standaryzacji, to transformacja korzystająca z rozkładu wartości w poszczególnych cechach:

$$a'_i = \frac{a_i - \bar{a}_i}{\sigma_i(x)}$$

Gdzie: $\bar{a}_i = \frac{1}{n} \sum_l a_{il}$ $\sigma_i(a) = \frac{1}{n-1} \sum_l (a_{il} - \bar{a}_i)^2$

Wynikiem transformacji jest wektor cech, gdzie wartość średnia tych cech $\bar{a}_i = 0$ i odchylenie standardowe $\sigma_i = 1$. Należy uważać by odchylenie standardowe nie było równe $\sigma_i = 0$, ponieważ spowoduje to dzielenie przez zero. Żeby temu zapobiec należy przed transformacją usunąć tzw. płaskie wzorce (ang. *flat pattern*).

1.4.4 Kalibracja klasyfikatora

W celu skorzystania z klasyfikatora należy go wytrenować. Do tego potrzebny jest dobór parametrów dla klasyfikatora oraz zbiór uczący. Jednym z algorytmów pomagających w dobieraniu parametrów jest algorytm przeszukiwania siatki (ang. *grid search*).

Działanie algorytmu polega na doborze każdego parametru z każdym możliwym do wyboru (kombinacji wszystkich parametrów). Każdą taką kombinację wartości parametrów nazywamy *węzłem*. Przyjmuje się że wartości parametrów są potęgami liczby 2.

Przykładowo w SVM wykorzystującym jądro radialne, dwoma najistotniejszymi parametrami są: C i γ .

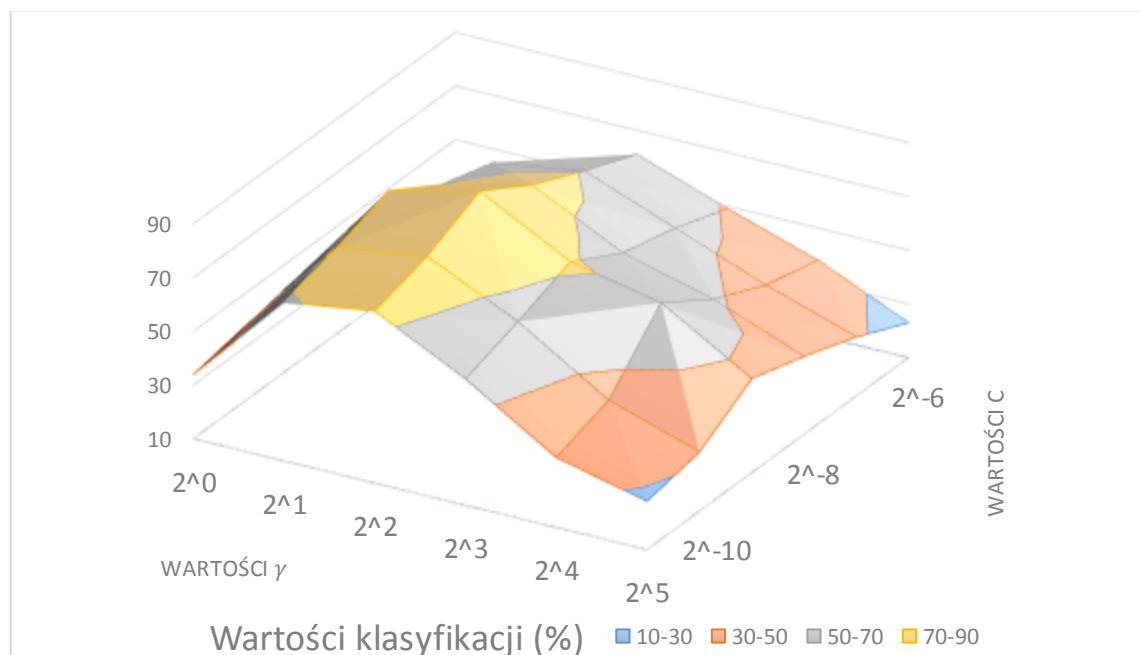
Działanie algorytmu zostało przedstawione poniżej w pseudokodzie:

1. Rozpocznij z pewnym zbiorem A_n , $i = 0$, gdzie n to liczba elementów w zbiorze A ;
2. Zbiór B_m , $j = 0$, gdzie m to liczba elementów w zbiorze B ;
3. $C = A_i$, $\gamma = B_j$;
4. Trening klasyfikatora z parametrami C oraz γ ;
5. Testowanie klasyfikatora i wypisanie wyniku;
6. $j++$;
7. Jeśli $j < m$, wróć do kroku 3.
8. $i++$;
9. Jeśli $i < n$, wróć do kroku 2.

Zakładając że:

- A zbiór wartości parametru C , $A = \{2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}\}$,
- B zbiór wartości parametru γ , $B = \{2^0, 2^1, 2^2, 2^3, 2^4, 2^5\}$.

Wyniki współczynnika klasyfikacji, wykorzystując algorytm przeszukiwania siatki, zostały przedstawione na Wykres 1.



Zaletą tego algorytmu jest jego prostota i możliwość wykonywania w sposób równoległy obliczeń z różnymi parametrami. Wadą jest znaczny nakład obliczeniowy. Obliczenia należy powtarzać tyle razy ile jest węzłów w naszej siatce.

1.4.5 Klasyfikacja

Klasyfikacja² jest to podział obiektów na grupy (klasy), według pewnych ściśle określonych zasad. Występuje wiele metod klasyfikacji obiektów. Najbardziej znane zostały opisane w rozdziale drugim opisane zostały najbardziej znane klasyfikatory.

Zbiór etykiet klas dla zadanego problemu klasyfikacyjnego będziemy oznaczać:

$$\mathbb{Y} = \{C_1, C_2, \dots, C_Y\},$$

gdzie Y , to liczba dostępnych etykiet klas. Najczęściej etykiety klas oznaczane są kolejnymi liczbami naturalnymi rozpoczynając od 0.

Oznaczając przestrzeń możliwych wartości wzorców przez \mathbb{X} oraz klasyfikator przez Ψ , mówimy że klasyfikator Ψ odwzorowuje przestrzeń cech \mathbb{X} w zbiorze etykiet \mathbb{Y} .

$$\Psi : \mathbb{X} \rightarrow \mathbb{Y}$$

Klasyfikacja to generowanie obszarów decyzyjnych (ang. *decision regions*) przez klasyfikator Ψ .

$$\bigwedge_{Y \in \mathbb{Y}} \{D_x^Y = (x \in \mathbb{X} : \Psi(x) = Y)\}$$

1.5 Weryfikacja

Rzeczą nieuniknioną jest popełnianie błędów przez systemy rozpoznawania. Może to być spowodowane klasyfikowaniem nieseparowanych klas lub dysponowaniem zbyt małym zbiorem treningowym. Są to przypadki kiedy nie jest możliwe uniknięcie błędów klasyfikacji. W przypadku gdy zadaniem klasyfikatora jest określenie przynależności próbki do jednej z dwóch etykiet klas (problem binarny), w statystyce na ocenę wyniku klasyfikatora definiuje się dwie miary:

- Czułość $S_n = \frac{TP}{TP+FN}$
- Specyficzność $S_p = \frac{TN}{TN+FP}$

² <http://sjp.pwn.pl/slowniki/klasyfikacja.html>

Gdzie:

- *TP* (ang. *true positive*) – prawidłowe wskazanie pierwszej klasy,
- *TN* (ang. *true negative*) – poprawne wskazanie drugiej klasy,
- *FP* (ang. *false positive*) – niepoprawne wskazanie pierwszej klasy,
- *FN* (ang. *false negative*) – niepoprawne wskazanie drugiej klasy.

Głównymi metodami sprawdzenia klasyfikacji są: metoda resubstytucji, metoda wydzielenia, metoda minus jednego elementu oraz metoda k-krotnej walidacji skróśnej.

- Metoda resubstytucji przyjmuje, że zbiorem uczącym jest ten sam zbiór co zbiór testowy. Metoda ta bardzo zawyża współczynnik klasyfikacji i często jest równy 100%.
- W metodzie wydzielenia, zbiór uczący jest wybierany w sposób losowy lub w inny określony sposób, gdzie reszta zbioru służy jako zbiór testowy. Przykładowo zbiór próbek zostaje podzielony równo, pierwsza połowa należy do zbioru uczącego a druga do zbioru testowego. Wadą jest zaniżanie współczynnika klasyfikacji, spowodowanej nie wykorzystaniem próbek z zbioru testowego do wyuczenia klasyfikatora.
- Metoda minus jednego elementu wykorzystuje zbiór N elementowy. Z tego zbioru zostaje usunięty jeden element, który zostaje zbiorem testowym. Reszta $N-1$ elementów należy do zbioru treningowego. Współczynnik klasyfikacji może być równy 0% lub 100%.
- Ostatnią metodą jest metoda k-krotnej walidacji skróśnej, polegająca na podziale zbioru na k podzbiorów zawierających taką samą (lub zbliżoną) ilość elementów. Następnie zbiór uczący składa się z $k-1$ podzbiorów, ostatni podzbiór zostaje zbiorem testowym. Proces jest powtarzany do momentu aż każdy z podzbiorów zostanie zbiorem testowym.

W celu dokładnego sprawdzenia skuteczności klasyfikatora konieczne jest kilkukrotne przeprowadzenie testu dla metod wydzielenia oraz k-krotnej walidacji skróśnej. W przypadku metody minus jednego elementu dobrze jest przeprowadzić ją tyle razy ile jest elementów w zbiorze (aby każdy element został zbiorem testowym).

Jeżeli wartość współczynnika klasyfikacji jest niska należy zastanowić się nad:

- zmianą klasyfikatora lub jego parametrów,
- zmianą cech,
- doбором innego zbioru treningowego (innymi próbkami, zwiększeniem zbioru treningowego).

2. Klasyfikatory

Angielskie określenie *methods pattern recognition* zwykle tłumaczone jest jako „metody rozpoznawania obrazów”, choć właściwszym wydaje się być określenie „metody rozpoznawania wzorców”, gdyż metody rozpoznawania obrazów są tylko jednym szczególnym przypadkiem metod rozpoznawania wzorców. Innymi metodami są rozpoznawanie: dźwięków, filmów, chorób, sytuacji ekonomicznej a nawet nastroju społeczeństwa.

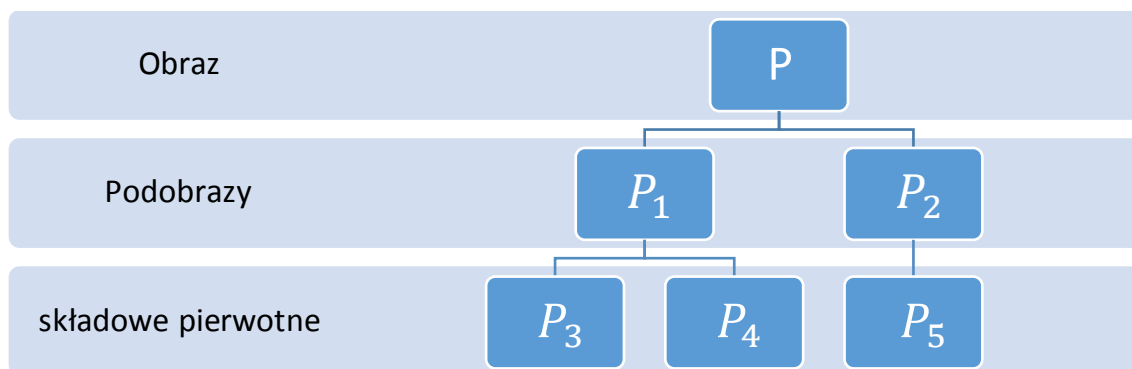
Detekcja obrazu jest dokonywana na różne sposoby w zależności od wybranej metody. Rozpoznawanie obrazu można podzielić na dwa główne sposoby, rozpoznawanie syntaktyczne i rozpoznawanie decyzyjno-teoretyczne [26].

2.1 Rozpoznawanie syntaktyczne

Rozpoznawanie syntaktyczne nazywane jest również lingwistycznym lub strukturalnym. Metoda powstała w połowie lat sześćdziesiątych, gdy okazało się, że przez metody matematyczne, nie uzyskuje się upragnionych celów w pewnych przypadkach. „Szczególnie dotyczyło to sytuacji, w których rozważane obrazy były bardzo złożone, lub liczba klas była bardzo duża, co z kolei powodowało znaczne zwiększenie się wymiaru przestrzeni cech użytych do opisu wzorców.” [1]. Tego typu problemy zostały rozwiązane podejściem syntaktycznym z wykorzystaniem metod lingwistyki matematycznej.

Metodę tę cechuje dzielenie obrazu na podobrazy, wykorzystującą zasadę dziel i rządź. Obraz jest dzielony do momentu, aż uda się rozpoznać obraz metodami całościowymi, traktując je jako niezależne. W przypadku bardzo złożonych obrazów, obraz jest dzielony aż do pierwotnych składowych obrazu.

Pierwotne składowe obrazu są to składowe niepodzielne, które istnieją oraz dla danej klasy obrazów potrafimy je zdefiniować. Do wyodrębnienia pierwotnych składowych najczęściej wykorzystujemy segmentację obrazu np. rozpoznawanie krawędzi za pomocą operandów dopasowania wzorca lub operandów gradientowych. Rysunek 7 przedstawia graficznie podział obrazu aż do pierwotnych składowych.



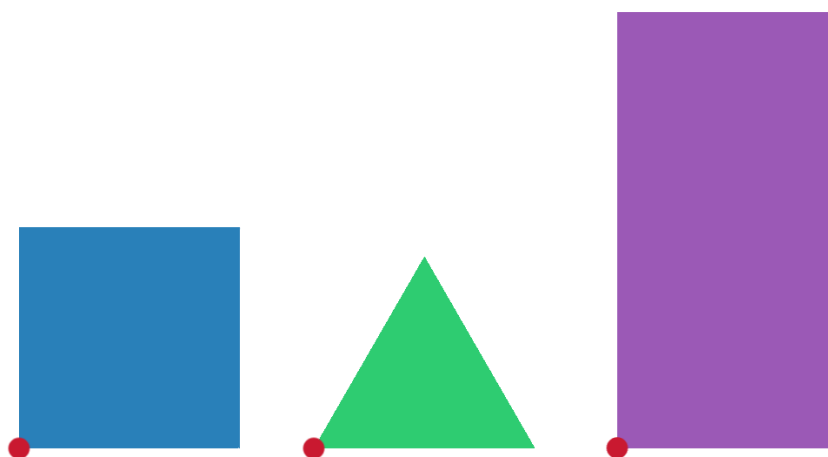
Rysunek 7. Graf dekompozycji obrazu na elementy składowe

2.1.1 Metody ciągowe

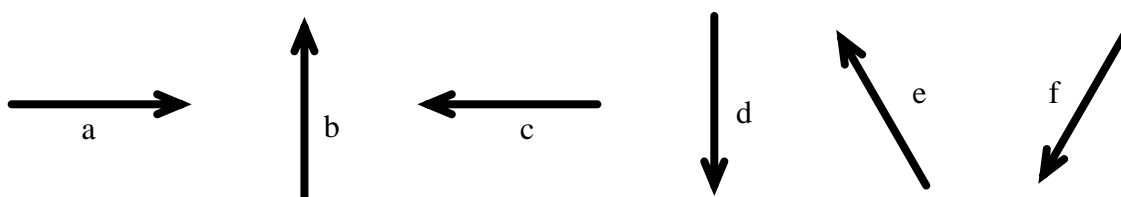
W podrozdziale 2.1 zostało wspomniane o metodach wyodrębniania pierwotnych składowych, jak również, że są one niezależne od siebie. Przy próbie złożenia obrazu do pierwotnej postaci, okazuje się to niemożliwe. Z pomocą przychodzą relacje pomiędzy składowymi pierwotnymi, zapewniające poprawną identyfikację relacji między podobrazami.

Pierwszą grupą metod są metody ciągowe, gdzie obraz jest reprezentowany przez ciąg a relacja jest stała (konkatencja). Biorąc zbiór obrazów przedstawiony na Rysunek 8 z założeniem, że zbiorem składowych pierwotnych jest Rysunek 9. Możemy ze zbioru obrazów opisać każdą obiekt za pomocą zbioru składowych pierwotnych.

Obraz pierwszy możemy opisać ciągiem $a^2b^2c^2d^2$, zaczynając od punktu w lewej dolnej krawędzi rogu opisując przeciwnie do kierunku wskazówek zegara. Drugi możemy opisać ciągiem $a^2e^2f^2$, postępując w ten sam sposób, co poprzednio. Ostatni możemy opisać ciągiem $a^2b^4c^2d^4$, w sposób identyczny jak poprzednio.



Rysunek 8. Zbiór obrazów poddany opisowi strukturalnemu



Rysunek 9. Zbiór składowych pierwotnych do opisu obrazów z rysunku wyżej

Głównym zastosowaniem języków ciągowych jest opisywanie pojedynczych, lecz często bardzo złożonych obiektów obrazu. Do opisywania obrazów zawierających więcej obiektów służą języki drzewowe i grafowe.

2.1.2 Metody drzewowe

Metody drzewowe służące do rozpoznawania obrazu bazują na automatach drzewowych oraz teorii języków drzewiastych. Automaty reprezentowane za pomocą drzewa są bardziej skomplikowane niż automaty ciągowe. Najczęstszym automatem służącym do rozpoznawania obrazu jest automat deterministyczny, rozpoczynający analizę struktury drzewa od liścia do korzenia.

Metody drzewiaste, które wykorzystują ten automat opierają się o drzewa takie jak EDT (ang. *Edge-labelled Directed Tree*) o skierowanych i zaetykietowanych krawędziach oraz drzewie T o nieskierowanych i niezaetykietowanych krawędziach.

Przykładem zastosowania tych drzew są aplikacje do analizy scen lub analizy faktur. Do analizy scen wykorzystuje się metody oparte o drzewo EDT a do analizy faktur metody oparte o drzewo T.

Pod pojęciem analizy scen rozumie się identyfikację i zidentyfikowanie obiektów na obrazie oraz opisanie relacji, zachodzących pomiędzy nimi. Pojęcia analizy faktur używa się do określenia powtarzającego się wzorca.

2.1.3 Metody grafowe

Kolejnymi metodami rozpoznawania syntaktycznego są metody grafowe. Różnią się one od metod wcześniej opisanych mocniejszym formalizmem. Co powoduje częstsze korzystanie z tych metod do opisu obrazów, dwuwymiarowych jak i trójwymiarowych.

Wykorzystywanie metod grafowych ma swoje ograniczenia w praktycznych rozwiązaniach systemów rozpoznawania obrazów. Spowodowane jest to dużą złożonością obliczeniową podczas analizy składniowej (ang. *parsingu*).

2.2 Rozpoznawanie decyzyjno-teoretyczne

Zupełnie innym podejściem od wcześniej opisanych metod jest rozpoznawanie decyzyjno-teoretyczne, znane również jako rozpoznawanie deterministyczne.

Rozpoznawanie decyzyjno-teoretyczne pozwala na rozwiązywanie zadań diagnostycznych przy wykorzystaniu rachunku prawdopodobieństwa oraz wielowymiarowej geometrycznej przestrzeni. Charakteryzuje się ono sposobem reprezentacji obrazu, poprzez pewien zbiór cech.

Rozpoznawanie obrazu (opisane w 1.4.5 Klasyfikacja) polega na podziale przestrzeni na klasy, za każdym razem przyporządkowując dany fragment badanego obrazu do jednej z klasy [4][26]. Tego typu rozpoznawanie dzielimy na klasyfikacje:

- Klasyfikacja nadzorowaną
- Klasyfikacja nienadzorowana

2.2.1 Klasyfikacja nienadzorowana

Klasyfikacją nienadzorowaną obrazu nazywamy procedurę, która dokonuje selekcji pikseli wybranego obrazu do poszczególnych klas obrazu. W jednej klasie powinny znajdować się piksele gdzie ich wartości są podobne lub zbliżone do siebie przy ustalonym zakresie lub zakresach kwalifikacyjnych. W pozostałych klasach wartości pikseli muszą być spoza wcześniej ustalonych zakresów kwalifikacji do klasy. Trzeba jednak pamiętać, że ilość klas nie powinna być duża.

Najczęściej podawanym przykładem w literaturze jest obraz po klasyfikacji zdjęcia satelitarne, który jest podzielony na klasy reprezentujące typy powierzchni takie jak obszar wodny, skalisty, obszar zabudowany itp. Satelita przy pewnych zakresach częstotliwości, rejestruje natężenie promieniowania różnych terenów o określonych wartościach. Następnie w prosty sposób klasy te mogą zostać skonwertowane do postaci wektorowej, natomiast obszary poszczególnych klas staną się wielobokami.

Algorytmy oparte na klasyfikacji nienadzorowanej cechuje brak informacji wstępnych na temat charakterystyki poszczególnych klas oraz ich zakresów kwalifikacyjnych. Można jednak ustalać inne ważne parametry takie jak: ilość dostępnych klas, maksymalnego dopuszczalnego wewnątrz jednej klasy odchylenia standardowego wartości pikseli itp. Jednym z algorytmów wykorzystujących klasyfikację nienadzorowaną jest algorytm k-średnich.

2.2.1.1 Algorytm k-średnich

Algorytm k-średnich jest najbardziej popularnym algorytmem iteracyjnym do klasyfikacji obrazu. Cechą charakterystyczną algorytmu jest konieczność wcześniejszego podania ilości klas obrazu.

Sposób działania algorytmu możemy podzielić na 5 kroków:

- Wybranie ilości dostępnych maksymalnie klas
- Ustalenie wstępnych początków centralnych. Często spotykaną metodą jest wybieranie losowych obiektów ze zbioru i uruchomienie programu kilkakrotnie, wybierając model wyników, który wydaje nam się najlepszy. W literaturze można również znaleźć informację aby, dobierać reprezentantów grup tak, by były od siebie jak najbardziej oddalone. Ma to usprawnić proces grupowania.
- Obliczenie odległości obiektów od środków skupień. Polega on na doborze odpowiedniej metryki, która ma za zadanie klasyfikację obiektów. Najczęściej używa się odległości euklidesowej, jej kwadratu lub odległości Czebyszewa.
- Przypisanie obiektu do skupienia, poprzez ustalenie do którego ma najbliżej.
- Ustalenie nowych środków skupień. Najczęściej wyznaczane są przez obliczenie średniej arytmetycznej współrzędnych obiektów należących do danego skupienia.

Proces powtarza kroki trzeci, czwarty i piąty, aż do spełnienia warunku zatrzymania. Najczęściej jest nim brak zmian w klasyfikacji obiektów do klas obrazu. Spotykana jest również z góry ustalona wartość ilości powtórzeń [5].

2.2.1.2 Algorytm ISODATA

Algorytm ISODATA (ang. *Iterative Self-Organizing Data Analysis Technique*) cechuje brak konieczności podawania wstępnej ilości klas obrazu. Dopuszczalne jest jednak podanie wartości minimalnych liczb obiektów w jednej grupie, podanie wartości maksymalnych rozrzutu grup, liczby ich łączonych oraz wartości oczekiwanej liczby skupień. Wartości te dobierane są metodą prób i błędów.

Sposób działania algorytmu jest bardzo podobny do algorytmu k-średnich:

- Wybranie (często losowo) początków centralnych.
- Dla każdego początku centralnego liczone jest odchylenie standardowe i odległości obiektów od środków skupień (tak samo jak w algorytmach k-średnich).
- Sprawdzane jest czy skupienie należy podzielić czy połączyć. Dzielimy grupę, gdy odchylenie standardowe skupienia jest większe od podanego przez użytkownika. Łączymy z sobą, gdy odległość między punktami centralnymi jest mniejsza od tej, którą podał użytkownik.
- Wyznaczenie punktu środka skupienia.

Algorytm powtarza krok drugi, trzeci i czwarty aż do momentu spełnienia warunku zakończenia. Warunkiem zakończenia algorytmu jest uzyskanie maksymalnej wartości iteracji, innymi przypadkami są nieprzekroczenie minimalnej wartości podanej przez użytkownika, średniej odległości między punktami centralnymi lub zmiany średniej odległości między punktami centralnymi [6].

2.2.2 Klasyfikacja nadzorowana

Proces klasyfikacji nadzorowanej jest częściowo zautomatyzowany, przez system komputerowy. Wynika to z procesu składającego się z dwóch etapów. W pierwszym etapie, osoba klasyfikująca podaje pola treningowe, czyli fragmenty obrazu będące wzorcami klas (budowanie modelu klasyfikatora). Ważne jest by wybrane wzorce klas obrazu były reprezentatywne.

Format modelu oraz informacje w nim zawarte różni się od wybranego klasyfikatora i może on przyjąć postać: drzewa decyzyjnego, reguł (if ... then ...), sieci neuronowych i inne. Ważnymi czynnikami wchodzącymi w skład oceny modelu klasyfikatora są: szybkość (czas uczenia się oraz klasyfikacji), odporność (szumy i błędne wartości), złożoność struktury (np. rozmiar drzewa decyzyjnego).

W drugim etapie system, sam klasyfikuje resztę pikseli, korzystając z wcześniej utworzonego modelu klasyfikatora.

Do klasyfikacji nadzorowanej możemy zaliczyć metody:

- Sieci neuronowe
- Maszyna wektorów nośnych
- Minimalno-odległościowe
- Drzewa decyzyjne
- Naiwny klasyfikator Bayesa

2.2.2.1 Sieci neuronowe

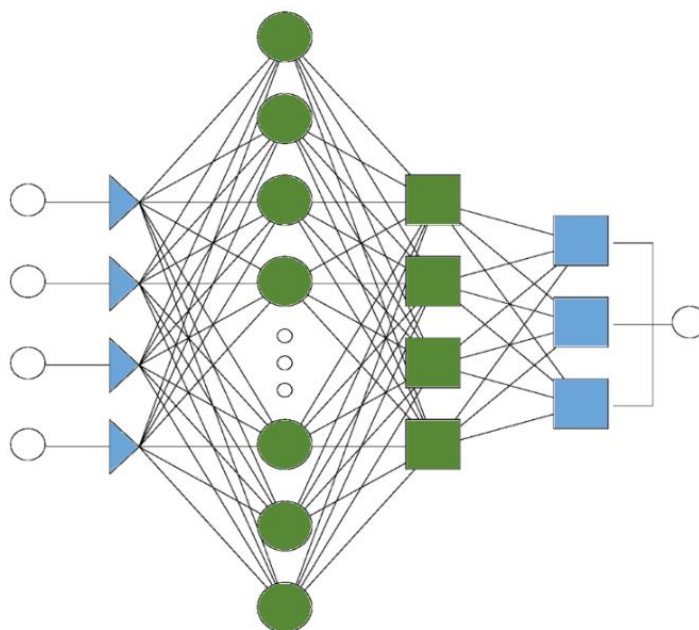
Metody wykorzystujące sieci neuronowe są najbardziej popularną techniką rozpoznawania obrazów. W roku 1989 zaprezentowano w pracy [3], pierwszy dobrze działający w pełni zautomatyzowany system rozpoznawania obrazów. Program opierał się na wielowarstwowej sieci neuronowej z systemem uczącym. Jego zadaniem było rozpoznawanie pisma drukowanego jak i ręcznego. Skuteczność tego systemu wyniosła w granicach pisma ręcznego 97% a pisma drukowanego 100%. Struktura systemów sieci neuronowych jest wzorowana w mniejszym lub większym stopniu na działaniu ludzkiego

mózgu i systemu neuronowego. Podstawową jednostką tego systemu są sztuczne neurony.

Mimo uproszczenia budowy neuronów, sieć neuronowa zachowuje wszystkie swoje cechy, które są ważne przy rozpoznawaniu zadań. Najczęściej spotykanymi strukturami sieci neuronowych są *feedforward* (tłum. „sieci z jednokierunkowym przepływem sygnału”) drugą spotykaną strukturą jest rekurencyjne. Różnica występująca pomiędzy strukturami tych sieci to kierunek przepływu informacji.

Struktura sieci rekurencyjnych cechuje się sprzężeniem zwrotnym. Co może powodować bardzo długie krążenie impulsów. W sieciach *feedforward* kierunek przepływu jest ściśle określony. Sygnał przepływa od pewnego ustalonego wejścia do określonego wyjścia. Ciekawe przykłady (budowy sieci) można znaleźć w [10].

Popularnym typem sieci neuronowych jest sieć z wielowarstwowym perceptronem (MLP). Sieć ta składa się z trzech warstw: wejściowej, wyjściowej oraz ukrytej, która znajduje się pomiędzy nimi. Neurony są połączone ze sobą jak na Rysunek 10.



Rysunek 10 Przykładowa struktura sieci neuronowej

Sygnał przychodzący do neuronu związany jest z pewną wagą wpływającą na normalizację informacji oraz ich modyfikację. Proces uczenia sieci neuronowej polega na doborze wag, często jest to wykonywane w sposób automatyczny na podstawie zbioru uczącego.

Dużą zaletą sieci neuronowych jest ich zdolność prawidłowego działania przy uszkodzeniu dużej części sieci neuronowej. Pokazuje to wielką przewagę jaką mają sieci neuronowe nad innymi metodami, które mogły już przestać funkcjonować przy dużym uszkodzeniu. Wadą sieci neuronowych jest długi i trudny proces uczenia (ustalanie odpowiednich wag).

2.2.2.2 Maszyna wektorów nośnych

Support Vector Machines przekładając na język polski oznacza maszynę wektorów nośnych. Nazwa ta jest jednak rzadko spotykana ze względu na małą ilość dostępnej literatury z tego tematu w języku polskim. Częściej można spotkać się ze skrótem SVM.

Metoda ta została wymyślona w 1995 roku przez Vladimira Vapnika [11]. Pozwala ona rozwiązać problemy klasyfikacji oraz regresji. Cechuje się skuteczniejszą klasyfikacją danych od innych klasyfikatorów, zwłaszcza gdy próbek w zbiorze uczącym jest mała ilość [12].

Zadaniem SVM jest wyznaczenie hiperpłaszczyzny w przestrzeni cech, która rozdzieli dane z największym marginesem [23]. Płaszczyzna dzieląca klasy nie zawsze jest liniowa, zwłaszcza gdy nie jest możliwe wyznaczenie gładkiej granicy pomiędzy klasami lub pomiędzy klasami występują szumy [33].

Jeśli możliwy jest podział liniowy przestrzeni, zadaniem klasyfikatora SVM jest wyznaczenie płaszczyzny H oraz płaszczyzn podpierających H_1 i H_2 . Płaszczyzny H_1 i H_2 muszą być równoległe i oddalone o tą samą długość od płaszczyzny H .

Zakładając:

\vec{x}_i - wektor cech opisujących próbkę,

$y_i \in \{-1, 1\}$ – etykieta klas,

\vec{w} - wektor normalny płaszczyzny.

Rozpatrując problem z dwoma klasami, dla zbioru par próbek $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$, płaszczyzny będą dane wzorami:

$$H: \vec{w} * \vec{x} - b = 0,$$

$$H_1: \vec{w} * \vec{x} - b = 1,$$

$$H_2: \vec{w} * \vec{x} - b = -1.$$

Odległość pomiędzy płaszczyznami H_1 i H_2 wynosi $\frac{2}{\|\vec{w}\|}$. W celu znalezienia najszerszego marginesu zmniejszamy wartość $\|\vec{w}\|$. Warunkami zapewniającymi, że pomiędzy płaszczyznami podpierającymi nie znajdzie się żaden obiekt są:

$$H_1: \vec{w} * \vec{x} - b = 1, \text{ dla } y_i = 1,$$

$$H_2: \vec{w} * \vec{x} - b = -1, \text{ dla } y_i = -1.$$

Warunkiem, że każdy punkt z klas znajduje się po właściwej stronie jest:

$$\bigwedge_{i \in \langle 1, n \rangle} \{ y_i (\vec{x}_i * \vec{w} - b) \geq 1 \}$$

W celu rozwiązania powyższego problemu oraz wyznaczenia równania hiperpłaszczyzny można wykorzystać mnożniki Lagrange'a [28].

Stosując funkcje jądrowe klasyfikator SVM może wyznaczać funkcje nieliniowe. Najczęściej stosowanymi funkcjami jądrowymi są:

- Liniowa
- Wielomianowa
- RBF (ang. *Radial Basis Kenel*)
- Gaussowska RBF
- Sigmoidalna

W [22] można znaleźć porównanie funkcji jądrowych.

2.2.2.3 Klasyfikatory minimalno-odległościowe

Najbardziej znanym klasyfikatorem typu minimalno-odległościowym jest k najbliższych sąsiadów, częściej oznaczane jako k-NN (ang. *K nearest neighbors*). Metoda polega na klasyfikacji obiektów do klas, wzorując się na przynależności K najbliższych sąsiadów. Stosując tą metodę odnosi się bardzo dobre wyniki, gdy liczba obserwacji jest bardzo duża.

Przed rozpoczęciem algorytmu następuje uczenie (stworzenie modelu klasyfikatora). Sposób działania algorytmu, można przedstawić w trzech krokach:

- Policzenie odległości pomiędzy wszystkimi wektorami zbioru treningowego a wektorem testowym. Odległości mogą być liczone na różne sposoby (Tabela 1).
- Przypisanie etykiety obserwowanego obiektu, który powtarza się najczęściej wśród k najbliższych sąsiadów. Jeżeli występuje równość między ilością etykiet między sąsiadami, problem rozwiązuje się losowo.

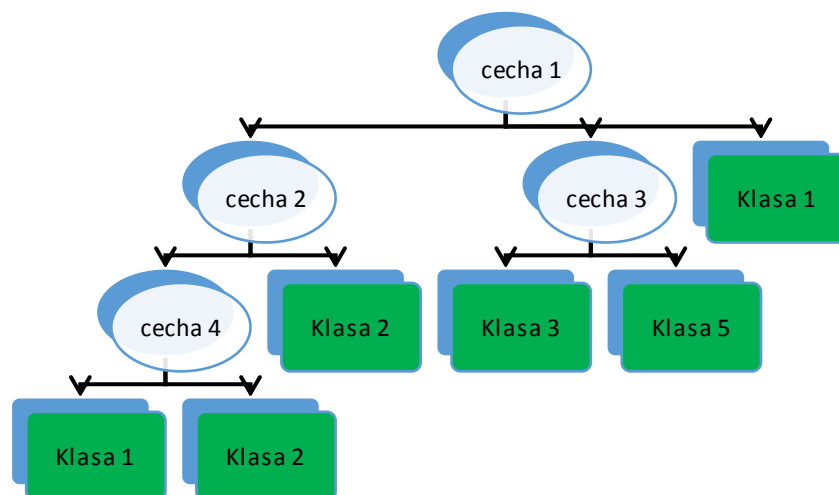
Nazwa odległości	Wzór
Euklidesowa	$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$
Canberra	$d(x, y) = \sum_i \left \frac{x_i - y_i}{x_i + y_i} \right $
Czebyszewa	$d(x, y) = \max_i (x_i - y_i)$
Manhattan	$d(x, y) = \sum_i x_i - y_i $

Tabela 1. Przykłady miar odległości.

2.2.2.4 Drzewa decyzyjne

Drzewa decyzyjne posiadają bardzo intuicyjną strukturę, składającą się z wierzchołków, które mogą być węzłem lub liściem. Każdy węzeł może posiadać jednego lub więcej potomków, połączonych gałęziami. Liście są węzłami, z których nie wychodzą żadne gałęzie oraz reprezentują etykietę klas.

Drzewa decyzyjne są acyklicznym grafem skierowanym. Budowane są od korzenia do liścia, w sposób rekurencyjny, metodą „dziel i zwyciężaj”. W taki sam sposób następuje klasyfikacja próbek.



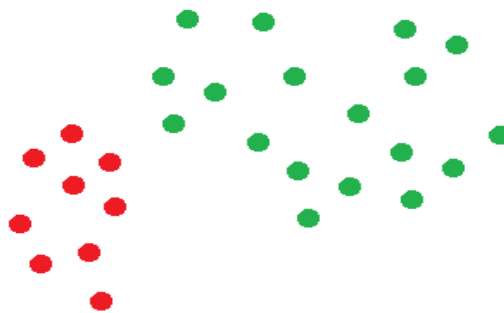
Jest wiele algorytmów tworzących strukturę drzewa decyzyjnego. Przykładowym jest znany algorytm iteracyjny dychotomizer (ID3), który został rozszerzony przez Rossa Quinlana [9][30].

Zaletą drzewa decyzyjnego jest szybkość klasyfikacji. Śledzenie procesu drzewa decyzyjnego jest zrozumiałe, jak również reprezentacja graficzna jest intuicyjna (dla człowieka). W łatwy sposób można prześledzić przejście próbki jak i zmodyfikować cechy. Dużą zaletą jest również tworzenie drzew w sposób rekurencyjny.

Do wad drzew decyzyjnych zaliczamy brak możliwości uczenia się adaptacyjnego jak również pojawienie się drastycznych błędów klasyfikacji w przypadku gdy blisko korzenia drzewa pojawi się błędna wartość atrybutu testowego.

2.2.2.5 Naiwny klasyfikator Bayesa

Naiwny klasyfikator Bayesa jest klasyfikatorem zaliczanym do podstawowych, prostych probabilistycznych klasyfikatorów. Nazwę naiwnego otrzymał z powodu swojego założenia wzajemnej niezależności predyktorów (zmiennych niezależnych), nie zawsze mającej związek z rzeczywistością. Klasyfikator sprawdza się dobrze w przypadku bardzo dużej ilości wymiarów na wejściu [32].



Rysunek 11 Ilustracja ułatwiająca wy tłumaczenie działania metody Bayesa. Przedstawia zielone i czerwone obiekty.

Przy założeniach że:

X – próbka, reprezentowana przez n -wymiarowy wektor, $X = (x_1, x_2, \dots, x_n)$

C_i - klasa

$P(C|X)$ – prawdopodobieństwo przynależności obiektu X do klasy C

$f(X|C_i)$ – gęstość rozkładu

$f(x)$ – suma gęstości wszystkich klas

Klasyfikacja nowego obiektu przypisuje próbkę do klasy która posiada największe prawdopodobieństwo.

$$P(C_i|X) = \frac{f(x|C_i)P(C_i)}{f(x)}$$

Rozpatrując przykład podany na rysunku powyżej (Rysunek 11), liczba obiektów zielonych jest dwa razy większa niż czerwonych. Z tego faktu wynika, że nowo klasyfikowany obiekt będzie miał dwa razy większe prawdopodobieństwo na zakwalifikowanie się do grupy zielonych. Takie prawdopodobieństwo nazywamy a priori.

Prawdopodobieństwo a priori obliczamy w następujący sposób:

$$\text{Dla zielonego: } P(C_1) = \frac{\text{ilość zielonych obiektów}}{\text{liczba wszystkich obiektów}}$$

$$\text{Dla czerwonych: } P(C_2) = \frac{\text{ilość czerwonych obiektów}}{\text{liczba wszystkich obiektów}}$$

Następną czynnością jest wyliczenie szansy przynależności obiektu:

Szansa że nowy obiekt zostanie zaklasyfikowany do obiektów zielonych:

$$P(X|C_1) = \frac{\text{ilość zielonych obiektów z pośród } k \text{ najbliższych sąsiadów próbki}}{\text{ilość wszystkich obiektów}}$$

Szansa że nowy obiekt zostanie zaklasyfikowany do obiektów czerwonych:

$$P(X|C_2) = \frac{\text{ilość czerwonych obiektów z pośród } k \text{ najbliższych sąsiadów próbki}}{\text{ilość wszystkich obiektów}}$$

Rezultatem klasyfikacji jest wybranie klasy o większym prawdopodobieństwie:

$$\prod_{j=1}^n (P(x_j|C_i)P(C_i))$$

gdzie n jest wielkością wektora reprezentującego próbkę, a x_j jest cechą próbki.

3. Biblioteka OpenCV

3.1 Instalacja biblioteki OpenCV

Biblioteka OpenCV jest dostępna w wielu systemach operacyjnych, korzysta się z niej w ten sam sposób na różnych systemach lecz inaczej przebiega proces instalacji.

3.1.1 Instalacji biblioteki na Linux

Instalacja na systemach Linux jest bardzo prosta ogranicza się do kilku komend. Przedstawiony proces instalacji nie zawiera komend, znajdują się one na oficjalnej stronie OpenCV.

- Instalacja odpowiednich narzędzi
- Sklonowanie repozytorium biblioteki do katalogu
- Zbudowanie projektu za pomocą programu CMake
- Zainstalowanie biblioteki

3.1.2 Instalacja na systemie Windows

Proces instalacji na systemie operacyjnym Microsoft Windows, biblioteki przebiega następująco:

- Pobranie pliku z oficjalnej strony producenta OpenCV
- Uruchomienie i wypakowanie
- Dodanie zmiennej środowiska do systemu oraz ścieżki systemowej

3.2 Moduł podstawowy

Rozpoczynając prace z biblioteką OpenCV należy poznać kilka podstawowych elementów tej biblioteki. Pierwszym i najważniejszym jest sposób reprezentacji zdjęcia.

3.2.1 Reprezentacja zdjęcia

Zdjęcie jest reprezentowane jako zbiór pikseli zapisanych w macierzy. W pamięci komputera jest to zbiór liniowy tej macierzy zapisany kolejno rzędami (zaczynając od górnego rzędu, a każdy rząd od lewej strony). W kodzie zdjęcie we wcześniejszych wersjach (języku C) było reprezentowane przez strukturę `IplImage`. Obecnie zdjęcie reprezentuje się przez strukturę (w języku C++) `Mat`.

Podstawowe operacje związane z strukturą Mat zostały podane w Tabeli 2.

Kod w OpenCV	Działanie operacji
Mat A = imread(lokalizacja obrazu)	Wczytanie obrazu do struktury Mat A.
Mat B = A.clone()	Sklonowanie macierzy A do macierzy B.
imwrite(lokalizacja obrazu, B)	Zapisanie obrazu na dysku.
A.at<uchar>(x, y)	Dostęp bezpośredni do piksela obrazu.

Tabela 2. Podstawowe operacje związane z strukturą Mat

3.2.2 Operacje na zdjęciu

Na zdjęciu za pomocą biblioteki OpenCV możliwe jest dokonanie wielu operacji. Do podstawowych operacji zalicza się:

- Skalowanie zdjęcia, zmieniające rozmiar zdjęcia w pionie lub w poziomie.
- Tworzenie masek, filtrów zwiększających np. ostrość zdjęcia.
- Mieszanie dwóch zdjęć, jest to nakładanie się dwóch zdjęć przy określonej przezroczystości.
- Zmiana kontrastu i jasności zdjęcia, gdzie częstym zastosowaniem jest poprawa jakości zdjęcia lub wyłuskanie interesującego nas obiektu.

Biblioteka OpenCV daje również możliwość rysowania prostych figur geometrycznych oraz wstawiania tekstów. Bardzo częstym zastosowaniem jest oznaczanie za pomocą okręgów lub prostokątów interesujących nas obiektów na obrazie.

3.3 Przetwarzanie obrazu

Zdjęcie nie zawsze przedstawia to co autor chciał pokazać, jak również interesujące nas obiekty nie są wystarczająco interpretowalne. Dlatego zdjęcie jest poddawane wcześniejszemu przetwarzaniu w celu ułatwienia interpretacji oraz analizy obrazu.

Biblioteka OpenCV posiada bardzo szeroki wachlarz narzędzi do tego procesu. Jednym z nich jest rozmycie. Jest to proces polegający na mieszaniu wartości pikseli z sąsiadującymi pikselami, opierający się na filtrach. Najczęściej wykorzystywanym filtrem jest Gaussowski. Można również użyć filtra znormalizowanego, medianowego lub dwustronny filtr. Gaussowski to typ filtru, gdzie każdy piksel przyjmuje wartość wyjściową z wartości jądra sąsiadujących pikseli, posiadających wagi wzorowane na krzywej Gaussa.

Dostępne rozmycia w bibliotece OpenCV zostały podane w Tabeli 3.

Kod w OpenCV	Typ rozmycia
Blur(A, B, Size(i, i), Point(-1, -1))	Rozmycie.
GaussianBlur(A, B, Size(i, i), 0, 0)	Rozmycie Gaussa.
medianBlur(A, B, i)	Rozmycie średnie.
bilateralFilter(A, B, i, i * 2, i / 2)	Filtr dwustronny.

Tabela 3. Rozmycia w bibliotece OpenCV

Zmienna	Wyjaśnienie
A	Macierz wejściowa, dostępne typy: CV_8U, CV_16U, CV_16S, CV_32F, CV_64F.
B	Macierz wyjściowa.
Size(i, i)	Wielkość jądra rozmycia.
Point(-1, -1)	Oznacza, że punkt kotwicy jest w centrum jądra.

Tabela 4. Wyjaśnienia do tabeli Tabela 3

3.3.1 Operacje morfologiczne

Kolejnymi metodami służącymi do poprawy interpretacji obiektów są operacje morfologiczne, gdzie podstawowymi przekształceniami są erozja oraz dyatacja, które zostały przedstawione w podrozdziale 1.2.4 Przekształcenia morfologiczne.

Przykłady zastosowania:

Kod w OpenCV	Działanie operacji
erode(A, B, E)	Przeprowadzenie operacji erozji na macierzy A.
dilate(A, B, E)	Przeprowadzenie operacji dyatacji na macierzy A.

Tabela 5. Podstawowe procesy morfologiczne

Zmienna	Wyjaśnienie
A	Macierz wejściowa, dostępne typy: CV_8U, CV_16U, CV_16S, CV_32F, CV_64F.
B	Macierz wyjściowa.
E	Element struktury wykorzystywany do erozji / dyatacji.

Tabela 6. Wyjaśnienia dla Tabeli 5

Do bardziej zaawansowanych operacji z rodziny metod morfologicznych należą metody otwarcia, domknięcia, gradientu, Top-Hat, Black-Hat.

Wszystkie wymienione operacje opierają się na operacjach erozji oraz dyatacji. Metoda otwarcia i domknięcia została wyjaśniona w podrozdziale 1.2.4 Przekształcenia morfologiczne.

Metoda gradientów jest różnicą metod dyatacji i erozji. Top-Hat jest równa różnicy zdjęcia wejściowego i zdjęcia po procesie otwarcia. Ostatnia metoda Black-Hat jest równa różnicy zdjęcia po procesie domknięcia i zdjęcia oryginalnego. Należy wspomnieć, że metoda gradientów może służyć do rozpoznawania konturów obiektów.

Przykład zastosowania:

Kod w OpenCV	Działanie operacji
morphologyEx(A, B, kindOfMorphology, E)	Zaawansowana przekształcenia morfologiczne

Tabela 7. Zaawansowane przekształcenia morfologiczne

Zmienna	Wyjaśnienie
A	Macierz wejściowa, dostępne typy: CV_8U, CV_16U, CV_16S, CV_32F, CV_64F.
B	Macierz wyjściowa.
kindOfMorphology	Typ operacji, dostępne opcje: 2 = otwarcia, 3 = domknięcia, 4 = gradientów, 5 = Top-Hat, 6 = Black-Hat.
E	Element struktury wykorzystywany do erozji / dyatacji.

Tabela 8. Wyjaśnienia dla Tabela 7

3.3.2 Krawędzie

Biblioteka zawiera także bardziej wyspecjalizowane metody w zakresie wykrywania krawędzi. Opierają się one na obliczaniu pochodnych. Pierwsza pochodna wyznacza szybkość zmiany wartości danej funkcji. W podobny sposób zostają wyszukane krawędzie. Są wyznaczane poprzez odnalezienie dużych wartości pierwszej pochodnej.

Za pomocą tej metody można wyznaczać kontury pionowe lub poziome. Rozwinięciem wyznaczającym wszystkie kontury pionowe i poziome jest operator Laplace'a. Opiera się na znalezieniu wartości zero dla drugiej pochodnej funkcji. Dokumentacja biblioteki podaje jeszcze kilka innych metod odpowiedzialnych za wyszukanie konturów obiektów na zdjęciach.

Przykład zastosowania:

Kod w OpenCV	Działanie operacji
Sobel(Z, A, ddepth, 1, 0, 3, scale, delta, BORDER_DEFAULT)	Znalezienie linii pionowych.
convertScaleAbs(A, abs_A)	Znalezienie linii poziomych.
Sobel(Z, B, ddepth, 0, 1, 3, scale, delta, BORDER_DEFAULT)	
convertScaleAbs(B, abs_B)	Połączenie znalezionych krawędzi pionowych i poziomych do macierzy C.
addWeighted(abs_A, 0.5, abs_B, 0.5, 0, C)	

Tabela 9. Metoda wykrywanie krawędzi (poziomych i pionowych)

Zmienna	Wyjaśnienie	
Z	Macierz wejściowa.	
A, B, abs_A, abs_B	Macierz wyjściowa.	
C	Macierz wynikowa.	
ddepth	Głębina obrazu wyjściowego:	
	Głębina macierzy wejściowej	Dostępne głębokości obrazu
	CV_8U	-1/CV_16S/CV_32F/CV_64F
	CV_16U/CV_16S	-1/CV_32F/CV_64F
	CV_32F	-1/CV_32F/CV_64F
	CV_64F	-1/CV_64F

Tabela 10. Wyjaśnienia dla Tabeli 9

3.3.3 Narożniki

Jedną z metod, sprawiających że maszyny mogą odnaleźć podobieństwo cech między dwoma obrazami jest określenie narożników obiektów na zdjęciu. Najbardziej popularna jest metoda wykrywania narożników Harris'a. Innymi metodami znajdowania narożników są: Shi-Tomasi, SIFT, SURF, położonych w subpikselach.

Przykład zastosowania:

Kod w OpenCV	Działanie operacji
<code>cornerHarris(Z, A, blockSize, apertureSize, k, BORDER_DEFAULT);</code>	Wykrywanie narożników metodą Harrisa.

Tabela 11. Metoda wykrywania narożników

Zmienna	Wyjaśnienie
Z	Macierz wejściowa w typie skali szarości lub zmiennoprzecinkowej.
A	Macierz wyjściowa.
blockSize	Wielkość sąsiedztwa.
apertureSize	Parametr przysłony.
k	Detektor Harrisa.

Tabela 12. Wyjaśnienia dla Tabeli 11

3.3.4 Przekształcenia

Jak w niemal każdym edytorze zdjęć tak i w bibliotece udostępnione jest kilka opcji pozwalających na łatwe przekształcenie zdjęcia. Jedną z takich funkcji jest dodawanie ramek, obracanie zdjęcia, przekształcenie w sposób odbicia lustrzanego pionowego, poziomego jak i przekształcenia afinicznego.

Przekształcenia afiniczne pozwalają na ujednolicenie obrazów. Przykładem takich przekształceń są przesunięcia, zmniejszenia, zwiększenia, obrót, pochylenia i przekształcenia panoramiczne.

Bardzo przydatną opcją jest skalowanie zdjęcia do innych rozmiarów. Pozwala nam ona na normalizację zdjęć w celu zachowania jednego rozmiaru dla wszystkich. Trzeba jednak pamiętać o stratach wynikających z tej operacji.

Przekształcenie polegające na wycięciu fragmentu obrazu można nazwać typem najprostszej segmentacji. Separacja ta opiera się o zmianę natężenia pomiędzy pikselami tła i interesującego nas obiektu.

Przykłady zastosowania:

Kod w OpenCV	Działanie operacji
<code>copyMakeBorder(A, B, top, bottom, left, right, borderType, value)</code>	Dodanie obramowania
<code>warpAffine(A, C, D, size)</code>	Obrót zdjęcia

Tabela 13. Dodanie obramowania

Zmienna	Wyjaśnienie
A	Macierz wejściowa.
B	Macierz wyjściowa z powiększoną szerokością o wartości <code>left + right</code> i wyższą o wartości <code>top + bottom</code> .
<code>top, bottom, left, right</code>	Wielkości ramki
<code>borderType</code>	Typ obramowania
Value	Wartość obramowania, jeśli <code>borderType</code> jest równy <code>BORDER_CONSTANT</code>
C	Macierz wyjściowa, z tym samym rozmiarem co macierz A
D	Macierz transformacji
Size	Wielkość macierzy wyjściowej

Tabela 14. Wyjaśnienia dla Tabeli 12

3.4 Interfejs użytkownika

Biblioteka OpenCV posiada wbudowaną bibliotekę HighGUI odpowiadającą za tworzenie interfejsu użytkownika.

Dużą zaletą jest to że ten sam kod bez modyfikacji działa na różnych systemach. Na przykład pisząc program na Windows będzie on działał również na Linux.

Interfejs, który można stworzyć jest ograniczony do obsługi okien, wyświetlania obrazów, obsługi sprzętowej (myszy i klawiatury), pobieraniu uchwytu okna i tworzenie suwaków oraz przycisków.

3.5 Histogram

Biblioteka poza ładowaniem zdjęcia oraz wyświetlaniem go oferuje możliwość stworzenia histogramu. Histogram jest to typ wykresu przedstawiający rozkład tonów w zdjęciu cyfrowym. Wykres jest tworzony za pomocą zliczania ilości pikseli dla każdego

tonu. Patrząc na histogram możliwe jest ocenienie rozkładu tonów, kontrastu, oraz stwierdzenie czy zdjęcie jest dostatecznie doświetlone itp.

3.6 Zaawansowane funkcje OpenCV

W podrozdziałach zostaną omówione bardziej złożone funkcje, które udostępnia biblioteka OpenCV. We wszystkich poniższych podrozdziałach uzyskiwanym efektem będzie film, możliwy do zapisu w dowolnym formacie.

3.6.1 Podstawowe operacje na filmach

Do podstawowych operacji należą odczyt i zapis filmu. Za odczyt filmu odpowiada klasa *cv::VideoCapture*, natomiast do zapisu wykorzystuje się klasę *cv::VideoWriter*. Odczyt filmu odbywa się klatka po klatce za pomocą operatora `>>`. Zapis posiada odwrócony operator `<<`. Konstruktor klasy *cv::VideoCapture* został przeładowany pozwalając na wykorzystanie kamery internetowej w komputerze. W celu jej uruchomienia w argumencie podajemy wartość 0.

3.6.2 Ekstrakcja

Ekstrakcja inaczej wyodrębnianie lub odejmowanie tła. Jest często stosowaną techniką generowania maski pierwszoplanowej (czyli binarnego obrazu zawierającego piksele należące do poruszającego się obiektu w scenie) z wykorzystaniem kamery statycznej. Jak sama nazwa wskazuje, odejmowanie tła polega na obliczeniu pierwszoplanowej maski wykonując odejmowanie bieżącej ramki i wzoru tła, zawierającą statyczną część sceny. Biblioteka OpenCV zawiera gotowy algorytm ekstrakcji zawarty w klasie *cv::BackgroundSubtractorMOG2*.

3.6.3 Rozpoznawanie twarzy

Klasyfikator kaskad *Haar* jest klasyfikatorem wykorzystywanym przez bibliotekę OpenCV do wykrywania twarzy. Przeprowadza on analizę sceny na każdym fragmencie obrazu, klasyfikując czy część obrazu zawiera twarz czy też nie. Klasyfikacja ta opiera się o zbiór uczący, który zawarty jest w bibliotece. Zbiór ten zapisany jest w pliku *haarcascade_frontalface_alt.xml*. W innych plikach zawarte są informacje do rozpoznawania oczu, ust, oraz innych części ciała człowieka.

4. Libsvm

Biblioteka libsvm jest implementacją klasyfikatora maszyny wektorów nośnych w wielu językach programowania: C++, java, python. Została skompilowana pod system Windows. W wyniku implementacji w tak dużej ilości języków możliwe jest skorzystanie z biblioteki na wielu platformach: Windows, Linux, Mac, Android, Windows Phone i innych.

Libsvm dzieli się na 3 główne programy: *predict*, *scale* i *train*.

- Po utworzeniu pliku zawierającego zbiór wektorów cech (którego format został podany poniżej), konieczne jest do poprawnej klasyfikacji przeskalowanie go. Do tego celu przeznaczony jest program *scale*. Sposób użycia:

```
svm-scale [opcje] nazwa_pliku_z_danymi.
```

Dostępne opcje:

-l dolna granica skalowania (domyślna wartość: -1),

-u górna granica skalowania (domyślna wartość: 1),

-y dolna i górna granica skalowania (domyślna wartość brak skalowania)

-s zapis parametrów skalowania do pliku

-r przywracanie skalowanych wektorów przy pomocy pliku z parametrami

Format pliku zbioru wektorów cech:

```
Numer_etykiety numer_cechy:wartość_cechy numer_cechy:wartość_cechy ...
```

```
:
```

```
Numer_etykiety numer_cechy:wartość_cechy numer_cechy:wartość_cechy ...
```

- Posiadając zbiór wektorów cech możliwe jest wyuczenie klasyfikatora, do tego można posłużyć się programem *train*.

Sposób użycia:

```
svm-train [opcje] nazwa_pliku_treningowego nazwa_pliku_klasyfikatora
```

Gdzie:

`nazwa_pliku_treningowego`: plik ze zbiorem przeskalowanych wektorów cech,

`nazwa_pliku_klasyfikatora`: plik wyuczonego klasyfikatora.

Najważniejsze opcje:

-s typ SVM (domyślnie 0 oznacza typ C-SVM),

-t jądro SVM (domyślnie 2 oznacza jądro RBF),

-d stopień w jądrze funkcji (domyślnie 3),

-g gamma (domyślnie 1/ilość cech),

-c współczynnik kary (domyślnie 1),

-e epsilon, tolerancja (domyślnie 0.001),

-v tryb k-krotnie walidacji,

-q tryb cichy (bez wypisywania wartości).

Poza wyuczeniem klasyfikatora, bardzo ważnymi funkcjami są walidacja pod parametrem `-v`, oraz możliwość zapisu wyuczonego klasyfikatora do pliku.

- Trzecim programem służącym do klasyfikowania jest program *predict*.

Sposób użycia:

`svm-predict [opcje] plik_testowy plik_ klasyfikatora plik_wynikowy`

Gdzie:

`plik_testowy`: plik ze zbiorem przeskalowanych wektorów cech,

`plik_ klasyfikatora`: plik wyuczonego klasyfikatora

`plik_wynikowy`: plik z wynikami

Dostępne opcje:

-b szacowanie probabilistyczne (domyślnie 0)

-q tryb cichy (bez wypisywania wartości).

Więcej informacji o bibliotece można znaleźć na stronie [37].

5. Przeprowadzenie eksperymentu

Eksperyment polegał na przeprowadzaniu procesów prowadzących do rozpoznania znaków tablic rejestracyjnych znajdujących się na zdjęciach. Celami tego eksperymentu było przedstawienie na przykładzie procesu rozpoznawania obrazu. W eksperymencie zostało zbadane czy możliwe jest przy wykorzystaniu sztucznie wygenerowanych znaków wyuczenie klasyfikatora w taki sposób aby klasyfikował on znaki ze zdjęć zrobionych aparatem w sposób poprawny.

Przed rozpoczęciem eksperymentu należało w poprawny sposób skonfigurować środowisko pracy. Do eksperymentu zostały wykorzystane Visual Studio 2015 oraz biblioteki OpenCV 3.1, QT i libsvm.

5.1 Przygotowanie zbioru danych

Początkiem eksperymentu było pobranie i zrobienie aparatem zdjęć, przedstawiających tablice rejestracyjne. W testach zostały wykorzystane dwa zbiory znajdujące się na płycie. Pierwszy zbiór składał się z tablic pobranych ze strony [31], który został pobrany przy pomocy skryptu napisanego w języku PHP. Skrypt ten został dołączony do płyty.

Działanie skryptu polega na wygenerowaniu ciągu znaków tablicy rejestracyjnej, przesłaniu formularza z danymi na stronę WWW [31], która generowała zdjęcie tablicy. Ostatnim krokiem skryptu była lokalizacja adresu zdjęcia w kodzie i pobranie go na dysk z nazwą numeru tablicy rejestracyjnej.

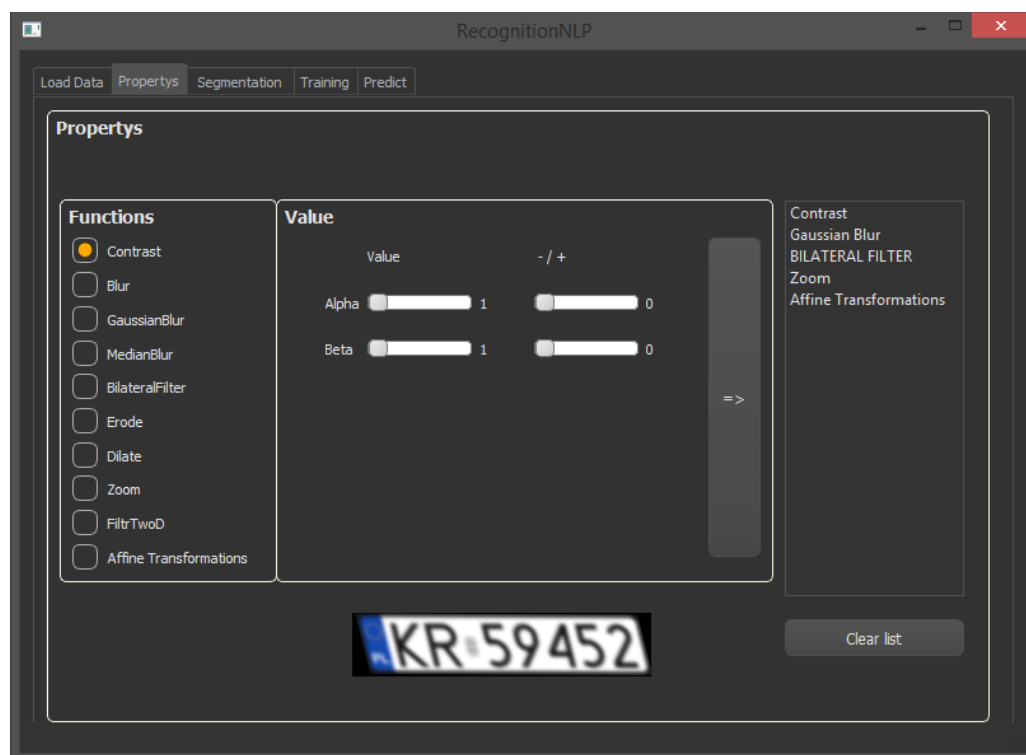


Rysunek 12 Przykład pobranej tablicy rejestracyjnej z strony [31]

W eksperymencie przyjąłem, że wszystkie tablice rejestracyjne rozpoczynają się literami KR lub KRA (tablice zarejestrowane w całym powiecie krakowskim). Łączny zbiór pobranych unikalnych tablic rejestracyjnych to 7 438. Wszystkie zdjęcia posiadają stały rozmiar, szerokość jest równa 235 pikseli a wysokość 50 pikseli. Dodatkowo zdjęcia są pozbawione wszystkich innych obiektów. Zdjęcie musi posiadać taką samą nazwę pliku jaką przedstawiają znaki na zdjęciu. Rozszerzenie pliku to *.png lub *.jpg.

Drugim zbiorem był zbiór tablic rejestracyjnych zrobionych aparatem przy pomocy smartfonu (Motorola XT1052). Łączna liczba wykonanych zdjęć jest równa 16.

Kolejnym etapem eksperymentu było zdeformowanie wszystkich obrazów wygenerowanych tablic (pobranych ze strony [31]). Przykładowy wynik deformacji przedstawia Rysunek 13.



Rysunek 13 Przedstawia przykładową deformację zdjęcia

Deformacja zdjęć odbywała się za pomocą programu dołączonego do płyty, przy pomocy przekształceń morfologicznych (opisanych w rozdziale 1.2.4) z wartościami domyślnymi dla programu. Wartość przekształceń to suma dwóch wartości. Pierwsza wartość znajduje się w kolumnie *Value* - oznacza wartość minimalną zniekształcenia. Drugą jest wartość w kolumnie *-/+* - oznacza wartość losową z przedziału od 0 do wybranej wartości.

Domyślnie wybranymi funkcjami są:

- Contrast (rozjaśniający zdjęcia $\text{Alpha} = 1$ z różnicą 0 oraz $\text{Beta} = 20$ z różnicą 20),
- Gaussian Blur (wygładzenie obrazu, $\text{sizeKernel} = 11$ z różnicą 5),
- Bilateral Filter (wygładzenie obrazu, $\text{sigmaColor} = 1$ z różnicą 30),
- Zoom (powiększenie oraz pomniejszenie dwukrotne).

5.2 Segmentacja

Po zdeformowaniu wszystkich wygenerowanych zdjęć, należało podzielić każde z nich tak, aby oddzielona część przedstawiała dokładnie jeden znak z tablicy rejestracyjnej.

Przed rozpoczęciem segmentacji należało jednak wykonać przetwarzanie wstępne, które w tym przypadku polegało na dokonaniu przekształceń na każdym obrazie. Zastosowane przekształcenia to: binaryzacja, filtr, przycięcie zdjęcia, negatyw i usuwanie fałszywych obiektów przy rogach obrazów.

Filtr i binaryzacja miały na celu wyróżnienie czarnych kolorów. Przycięcie zdjęcia służyło do usunięcia obramowania tablicy, które również było czarne. Usuwanie fałszywych obiektów miało jedynie zastosowanie przy zdjęciach robionych aparatem.

Segmentacja polegała na liczeniu średniej wartości wszystkich punktów w danej kolumnie zaczynając od lewej strony. Gdy wartość średnia całej kolumny przekraczała wartość progową, oznaczało to że w danej kolumnie znajduje się znak. Gdy wartość średnia całej kolumny nie przekraczała wartości progowej, znaczyło to że w danej kolumnie znaku nie ma.

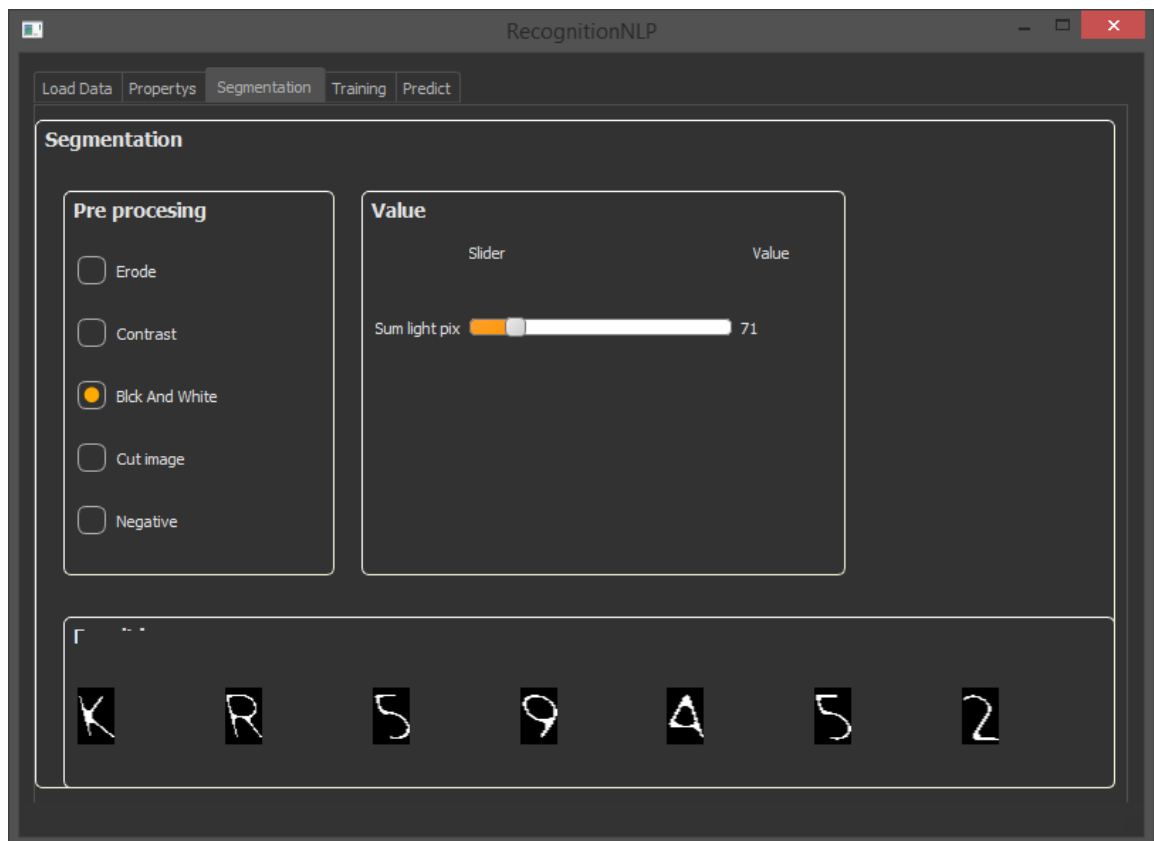
Wynika z tego, że początek znaku znajdował się w miejscu gdzie wartość średnia kolumny była mniejsza od wartości progowej a następna wartość średnia była większa od wartości progowej.

Koniec znaku był przeciwieństwem i znajdował się gdy wartość średnia kolumny była mniejsza od wartości progowej a poprzedzająca kolumna miała wartość średnią większą od wartości progowej. Po zlokalizowaniu znaku został on wycięty. Proces był powtarzany aż do znalezienia wszystkich znaków.

W zakładce *segmentation* możliwy jest podgląd w jaki sposób obraz będzie podzielony, co przedstawia Rysunek 14.

Domyślnymi wartościami są:

- Erode (pogrubienie krawędzi, $size = 0$, $kernel = 0$),
- Contrast (zmiana kontrastu, $Alpha = 0$, $Beta = 0$),
- Black And White (Binaryzacja zdjęcia. Kryterium jest jasność piksela),
- Cut image (Przycięcie zdjęcia, $left = 20$, $top = 5$, $right = 0$, $bottom = 5$),
- Negative (Negacja pikseli, domyślnie włączona).



Rysunek 14. Przykładowa wynik segmentacji

Po wyselekcjonowaniu wszystkich znaków ze zbioru wygenerowanych tablic rejestracyjnych, znaki zostały podzielone na dwa zbiory. Pierwszym był zbiór uczący, a drugim zbiór testowy.

Wszystkie obrazy zostały dodatkowo przekształcone geometrycznie. Przekształcenie polegało na obrocie o kąt nie większy (losowo) niż 20 stopni. Dodatkowo obrazy należące do zbioru uczącego zostały przekształcone również przez przesunięcie znaku w górę, dół, prawo, lewo o wartości od 0 do 5 pikseli losowo.



Rysunek 15. Znak pochodzący ze zdeformowanego, po przekształceniach geometrycznych zdjęcia należącego do zbioru uczącego



Rysunek 16. Znak pochodzący ze zdeformowanego zdjęcia należącego do zbioru testowego



Rysunek 17. Znak pochodzący ze zdjęcia wykonanego aparatem

5.3 Nauka klasyfikatora

Bardzo ważnym elementem było dobranie odpowiedniego klasyfikatora do problemu. Z powodu szybkości działania, radzenia sobie z wielowymiarowymi danymi oraz z powodu poprawności klasyfikacji przy małej ilości próbek w zbiorze uczącym, został wybrany klasyfikator SVM. Wykorzystałem bibliotekę libsvm z zaimplementowanym klasyfikatorem SVM. Testy zostały przeprowadzane na typie C_SVM z jądrami: liniowym oraz radialną funkcją bazową.

Kolejnym krokiem po segmentacji było wyuczenie klasyfikatora. Nie było jednak możliwe użycie surowych zdjęć, dlatego ze zdjęć zostały wyliczone wektory cech, składające się z momentów oraz momentów H_u , które zostały zapisane do pliku.

Kolejnym etapem było skalowanie wektorów przy pomocy biblioteki libsvm. Skalowane wektory cech tworzące zbiór uczący zostały zapisane w pliku VF_Training.txt, zaś zbiór testowy do pliku VF_Test_I.txt.

Ostatnim krokiem jaki należało wykonać aby wyuczyć klasyfikator był dobór parametrów klasyfikatora. Dobór odpowiednich parametrów nie był prosty, dlatego wykorzystany został skrypt, którego zadaniem było wygenerowanie wszystkich kombinacji wartości parametrów i sprawdzenia, który z nich jest najlepszy. Taki algorytm nazywamy przeszukiwaniem siatki, który został przedstawiony w podrozdziale 1.4.4.

Wszystkie kroki, mające na celu przeprowadzenie eksperymentu zostały uproszczone do przygotowania folderów ze zbiorami programu libsvm oraz kliknięcia przycisku *test* w stworzonym programie. W głównym katalogu programu powinny znajdować się dwa foldery: *SET* oraz *SCRIPTS*.

Drzewo poniżej przedstawia jak powinny zostać utworzone foldery oraz co powinno się w nich znajdować:

- Główny katalog: zawiera program
 - ↳ **SCRIPTS**: zawiera skrypty wyświetlające wyniki klasyfikacji
 - ↳ **Libsvm**: zawiera bibliotekę libsvm
 - ↳ **Classifier**: będzie przechowywać wektory cech testów oraz modele wyuczonych klasyfikatorów.
 - ↳ **SET**: tylko foldery
 - ↳ **NLP**: zawiera zbiór wygenerowanych zdjęć, przedstawiających tablice rejestracyjne.
 - ↳ **NLP_Camera**: zawiera zbiór zdjęć zrobionych aparatem, przedstawiających tablice rejestracyjne.
 - ↳ **Test_I**: będzie przechowywać zdjęcia przedstawiające znaki wykorzystane do testu pierwszego.
 - ↳ **Test_II**: będzie przechowywać zdjęcia przedstawiające znaki wykorzystane do testu drugiego.
 - ↳ **Training**: będzie przechowywać zdjęcia przedstawiające znaki wykorzystane do treningu klasyfikatora.

Po przygotowaniu odpowiednich folderów oraz zbiorów, należy włączyć przygotowany program i kliknąć na przycisk *test*. Po zakończeniu testu w folderze *SCRIPTS* zostaną wygenerowane pliki (które pomogą w znalezieniu optymalnych parametrów dla klasyfikatora, oraz przeprowadzą klasyfikację):

- **GS_Linear.bat**: wypisuje wartości współczynnika klasyfikacji dla dwóch testów wykorzystując jądro liniowe,
- **GS_RBF.bat**: wypisuje wartości współczynnika klasyfikacji dla dwóch testów wykorzystując jądro RBF,
- **TEST I.bat**: wypisuje wartości współczynnika klasyfikacji dla testu pierwszego z dwoma jądrami,
- **TEST II.bat**: wypisuje wartości współczynnika klasyfikacji dla testu drugiego z dwoma jądrami.

Oraz w folderze *SCRIPTS/Classifier* zostaną wygenerowane pliki oznaczające:

- „**VF_***”: oznaczają wektory cech przed normalizacją (testu I, II oraz zbioru uczącego),
- „**NVF_***”: oznaczają znormalizowane wektory cech (testu I, II oraz zbioru uczącego),
- „**Model_***”: oznaczają modele wyuczonych klasyfikatorów (ostatnio uruchomionego treningu klasyfikatora).

5.4 Wyniki eksperymentu

Do obu testów klasyfikatory zostały wyuczone przez zbiór uczący składający się z 100 próbek dla każdej z klasy (łącznie 3 500 obrazów). Wartości parametrów potrzebne do treningu klasyfikatora zostały podane poniżej (Tabela 15).

Typ SVM	Typ jądra	Parametry
C-SVM	Liniowy	$C = 1$
C-SVM	RBF	$C = 1, g = 0.128$

Tabela 15. Dobór parametrów dla klasyfikatorów

W pierwszym teście, zbiór testowy składał się z około 48 000 przygotowanych wcześniej znaków. Wyniki testu podane zostały poniżej (Tabela 16).

Typ SVM	Typ jądra	Współczynnik klasyfikacji
C-SVM	Liniowy	99,9188%
C-SVM	RBF	99,9339%

Tabela 16. Wyniki klasyfikacji dla zbioru wygenerowanego

W drugim teście, zbiór testowy składający się z około 110 znaków (pochodzących ze zdjęć wykonanych smartfonem). Wyniki testu zostały podane w tabeli poniżej (Tabela 17).

Typ SVM	Typ jądra	Współczynnik klasyfikacji
C-SVM	Liniowy	71,4286%
C-SVM	RBF	78,5714%

Tabela 17. Wyniki klasyfikacji dla zbioru zrobionego aparatem

6. Podsumowanie

W niniejszej pracy zostały krótko przedstawione metody rozpoznawania obrazów, zostały opisane narzędzia, którymi można się posłużyć do stworzenia systemu analizy obrazu statycznego jak i dynamicznego, co zostało również zaprezentowane w eksperymencie.

W rozdziale 1 zostało krótko zaprezentowane etapy metod rozpoznawania obrazów. Przedstawione zostały: wady i zalety najczęściej wykorzystywanych urządzeń do akwizycji obrazu, wykorzystywane funkcje podczas przetwarzania wstępnego, metody segmentacji i sprawdzające współczynniki klasyfikacji oraz znane z literatury algorytmy klasyfikacji.

W kolejnych rozdziałach zostały przedstawione biblioteki OpenCV oraz libsvm. Szczególną uwagę poświęcono możliwościom jakie oferuje biblioteka OpenCV, która jest często wykorzystywana przez systemy analizy obrazów.

Ostatni rozdział został poświęcony na część praktyczną, przedstawiającą eksperyment polegający na rozpoznawaniu znaków ze zdjęć przedstawiających tablice rejestracyjne. Rozdziałowi temu została poświęcona szczególna uwaga przedstawiającego wszystkie etapy rozpoznawania obrazów. Do wykonania eksperymentu został stworzony program wykorzystujący biblioteki Qt (dzięki której został stworzony interfejs użytkownika), biblioteki OpenCV (służącą do realizacji wszystkich operacji na obrazach) oraz libsvm, która posłużyła do celu użycia klasyfikatora SVM. Cały program został zaimplementowany w języku C++, łącznie zajmuje około 4 000 linii kodu.

Podczas implementacji programu pojawiało się dużo trudności związanych z łączeniem bibliotek do jednego projektu, zmiany interfejsu użytkownika, wyborem klasyfikatora i inne. Jednym z problemów był błąd wynikający ze złego zapisu nazw plików (tablic rejestracyjnych wygenerowanych) przez skrypt (napisany w języku PHP). Skrypt zapisywał nazwę pliku w sposób niepoprawny podczas gdy wykonywało się kilka tych samych skryptów w tym samym czasie. Problem został rozwiązany, poprzez uruchomienie tylko jednego skryptu pobierającego zbiór w tym samym czasie.

Eksperyment przedstawiony w pracy pokazuje, że współczynnik klasyfikacji klasyfikatora SVM, wykorzystującego sztucznie wygenerowane znaki, z których zostały wyliczone momenty geometryczne do zbioru uczącego, przekracza wartość 70%. Otrzymane wartości jakości klasyfikacji można zwiększyć wyliczając inne cechy z obiektów, które zostały przedstawione w [24][25]. Drugim powodem dla którego współczynnik klasyfikacji nie jest większy jest stworzony program który działa na sztywnych (ustalonych przez użytkownika) wartościach. Rezultatem jest niedopasowanie parametrów programu dla wszystkich przypadków. Eksperyment pokazuje również, że momenty geometryczne są odporne na obrót, przesunięcia oraz skalowania obiektów.

7. Bibliografia

- [1] - R. Tadeusiewicz M. Flasiński , Rozpoznawanie obrazów, Państwowe Wydawnictwo Naukowe Warszawa 1991, s. 122
- [2] - J. Koronacki, J. Ćwik, Statystyczne systemy uczące się, Wydawnictwo Naukowo-Techniczne Warszawa 2005
- [3] - Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, Backpropagation Applied to Handwritten, Zip Code Recognition 1989, s. 541-551
- [4] - R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification
- [5] - J. A. Hartigan and M. A. Wong, K-Means Clustering Algorithm 1979, Applied Statistics, 28, 100.
- [6] - A. R. Webb Statistical Pattern Recognition, 2nd Edition
- [7] - J. Koronacki, J. Ćwik, Statystyczne systemy uczące się, Wydawnictwo Naukowo-Techniczne Warszawa 2005
- [8] - Wachsmuth E.D., Becker M., Grieve A.P., Maurer-Schultze B., Fully automated and fast image analysis of autoradiographs with a TAS-Leitz. Determination of size, Feulgen fluorescence and grain counts of individual nuclei and their evaluation by a simplified cluster analysis, Histochemistry 1984, s. 227-236
- [9] - J.R. Quinlan, Discovering rules by induction from large collections of examples, in Expert Systems in the Micro Electronic Age, Edinburgh University Press, s. 168–201, 1979.
- [10] - W. Duch, N. Jankowski, Survey of neural transfer functions, Neural Computing Surveys 2, s. 163–212, 1999.
- [11] - V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995
- [12] - S. Abe, Support Vector Machines for Pattern Classification, Springer Verlag, New York, 2010.
- [13] - R. Tadeusiewicz, Odkrywanie sieci neuronowych przy użyciu programów w języku C#, Polska Akademia Umiejętności Kraków 2007

- [14] - R. Tadeusiewicz, P. Korohoda, Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo fundacji postępu telekomunikacji, Kraków 1997, s. 243-246
- [15] - M. Skomorowski, Wybrane zagadnienia rozpoznawania obrazów, Wydawnictwo Uniwersytet Jagielloński, Kraków 2013, s. 7-13
- [16] – W. Chmielnicki,
Efektywne metody selekcji cech i rozwiązywania problemu wieloklasowego w nadzorowanej klasyfikacji danych, Polska Akademia Nauk 2012, s. 44-58
- [17] – I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, „Journal of Machine Learning Research 3” 2003, s.1157-1182
- [18] – J. Piątkowski, Analiza i rozwój metod selekcji cech dla dużych problemów klasyfikacyjnych, Uniwersytet Mikołaja Kopernika w Toruniu 2003, s. 8-19
- [19] – R. Kohavi, J. George, Wrappers for feature subset selection, „Artificial Intelligence 97” 1997, s. 284-286
- [20] – M. Naseriparsa, Amir-Masoud Bidgoli, A Hybrid Feature Selection Method to Improve Performance of a Group of Classification Algorithms, “International Journal of Computer Applications Volume 69- No. 17” 2013, s. 28-31
- [21] – N. Lal, O. Chapelle, J. Weston,
Elisseeff Andre, Embedded Methods, s. 1-2
- [22] – M. Hussain, S. Wajid, A. Elzaart,
A Comparison of SVM Kernel Functions for Breast Cancer Detection, Riyadh 2011
- [23] – M. Orchel, Klasyfikacja danych wielowymiarowych algorytmami SVM, Akademia Górniczo hutnicza w Krakowie 2005 s. 8-14
- [24] – D. Xu, H. Li,
Geometric moment invariants, Pattern Recognition 41, 2008, s. 240–249
- [25] – R. Mukundan, K.R. RamaKrishnan, Fast Computation of Legendre and Zernike Moments, Pattern Recognition, s. 1433–1442.
- [26] – Podział metod klasyfikatorów [Online] [Ostatni dostęp: 04.05.2016]
http://luk.kis.p.lodz.pl/IT/Rozpoznawanie_obrazow_v1.1.pdf

- [27] - Algorytmy [Online] [Ostatni dostęp: 04.05.2016]
http://ue.poznan.pl/www/data/upload/image/katedra%20Statystyki/was_wyklad_6_marca_2013_prezentacja.pdf
- [28] - SVM [Online] [Ostatni dostęp: 04.05.2016]
<http://mst.mimuw.edu.pl/lecture.php?lecture=syd&part=Ch10>
- [29] - Algorytm K-NN [Online] [Ostatni dostęp: 04.05.2016]
http://www.tomaszx.pl/materialy/si_lab7.pdf
- [30] - Protein Data Bank [Online] [Ostatni dostęp: 04.05.2016]
<http://www.rcsb.org/pdb/home/home.do>
- [31] - Zbiór tablic rejestracyjnych [Online] [Ostatni dostęp: 04.05.2016]
<http://platesmania.com>
- [32] - Algorytm Bayesa [Online] [Ostatni dostęp: 04.05.2016]
http://www.statsoft.pl/textbook/stathome_stat.html?http%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstnaiveb.html
- [33] – J. Stefanowski, SVM [Online] [Ostatni dostęp: 13.05.2016]
Politechnika Poznańska <http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf> s. 4-44
- [34] – Podział cech [Online] [Ostatni dostęp: 15.05.2016]
http://eletel.p.lodz.pl/pstrumil/po/opis_cech.pdf
- [35] – M. Błachnik, Metody selekcji cech [Online] [Ostatni dostęp: 15.04.2016]
http://www.mblachnik.pl/lib/exe/fetch.php/dydaktyka/zajecia/wyklady/metody_selekcji_cech.pdf
- [36] – M. Shardlow,
An Analysis of Feature Selection Techniques [Online] [Ostatni dostęp: 15.04.2016]
<https://studentnet.cs.manchester.ac.uk/pgt/COMP61011/goodProjects/Shardlow.pdf>, s. 1-4
- [37] – Biblioteka libsvm [Online] [Ostatni dostęp: 04.05.2016]
<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [38] –P. Sturmiłło, Podział cech [Online] [Ostatni dostęp: 04.05.2016]
<http://eletel.p.lodz.pl/pstrumil/po/geometria.pdf>