

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA AUTOMATYKI I INŻYNIERII BIOMEDYCZNEJ



PRACA INŻYNIERSKA

RAFAŁ STUDNICKI

**ANALIZA WYBRANYCH DESKRYPTORÓW CECH
WYKORZYSTYWANYCH W ALGORYTMACH DETEKCJI
OBIEKTÓW**

PROMOTOR:

dr hab. inż. Marek Gorgoń,
prof. nz. AGH

Kraków 2012

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

.....

PODPIS

AGH
University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical
Engineering

DEPARTMENT OF AUTOMATICS AND BIOMEDICAL ENGINEERING



BACHELOR OF SCIENCE THESIS

RAFAŁ STUDNICKI

**ANALYSIS OF FEATURE DESCRIPTORS USED IN OBJECT
DETECTION ALGORITHMS**

SUPERVISOR:
Marek Gorgoń Ph.D Eng.

Krakow 2012

Miejsce na podziękowania

Spis treści

1. Wprowadzenie	8
1.1. Wykorzystanie komputerowych systemów wizyjnych	8
1.2. Znaczenie detekcji obiektów w komputerowych systemach wizyjnych	10
1.3. Rola deskryptorów cech w rozpoznawaniu obrazów	10
1.4. Cele pracy	11
1.5. Zawartość pracy	11
2. Dotychczasowe badania	12
2.1. Proste deskryptory	12
2.1.1. HOG	12
2.1.2. Kowariancja cech	13
2.1.3. LBP	14
2.1.4. Edgelets	14
2.2. Złożone deskryptory	15
2.3. Podsumowanie	16
3. Metodologia przeprowadzonych badań	17
3.1. Implementacja	17
3.2. Wykorzystywany zestaw danych	17
3.3. Klasyfikator - Support Vector Machine	18
3.4. Podział zbioru danych	19
3.5. Trening i walidacja	20
3.6. Dotrenowanie klasyfikatora	21
3.7. Detekcja w przesuwym oknie	21
3.8. Skalowanie obrazu testowanego	22
3.9. Eliminacja redundantnych detekcji	23
3.10. Walidacja z wykorzystaniem danych rzeczywistych	24
3.11. Sposób bezpośredniego porównania metod	27
3.12. Analiza złożoności metod	27
3.13. Podsumowanie	28

4. Wybrane deskryptory cech	30
4.1. HOG - Histogram of Oriented Gradients	30
4.1.1. Algorytm obliczenia deskryptora.....	30
4.1.2. Wyniki pierwszej walidacji	30
4.1.3. Przykłady fałszywych detekcji.....	31
4.1.4. Wyniki drugiej walidacji	32
4.1.5. Rozkłady błędów detekcji.....	32
4.1.6. Analiza złożoności poszczególnych kroków metody	33
4.1.7. Podsumowanie	34
4.2. Macierz kowariancji cech	35
4.2.1. Algorytm obliczenia deskryptora.....	35
4.2.2. Wyniki pierwszej walidacji	35
4.2.3. Przykłady fałszywych detekcji.....	36
4.2.4. Wyniki drugiej walidacji	37
4.2.5. Rozkłady błędów detekcji.....	37
4.2.6. Analiza złożoności poszczególnych kroków metody	38
4.2.7. Podsumowanie	39
4.3. LBP - Local Binary Patterns.....	40
4.3.1. Wprowadzenie.....	40
4.3.2. Algorytm obliczenia deskryptora.....	40
4.3.3. Wyniki pierwszej walidacji	41
4.3.4. Przykłady fałszywych detekcji.....	41
4.3.5. Wyniki drugiej walidacji	42
4.3.6. Rozkłady błędów detekcji.....	42
4.3.7. Analiza złożoności poszczególnych kroków metody	43
4.3.8. Podsumowanie	43
4.4. Edgelets	45
4.4.1. Wprowadzenie.....	45
4.4.2. Algorytm obliczenia deskryptora.....	45
4.4.3. Wyniki pierwszej walidacji	46
4.4.4. Analiza złożoności poszczególnych kroków metody	46
4.4.5. Podsumowanie	47
4.5. Zestawienie przebadanych metod.....	47
5. Łączenie działania deskryptorów	48
5.1. Wprowadzenie	48
5.2. Łączenie deskryptora kowariancji i LBP.....	48

5.2.1.	Wyniki pierwszej walidacji	48
5.2.2.	Wyniki drugiej walidacji	49
5.2.3.	Rozkłady błędów detekcji	50
5.2.4.	Analiza złożoności poszczególnych kroków metody	50
5.2.5.	Podsumowanie	51
5.3.	Łączenie deskryptora HOG i LBP	51
5.3.1.	Wyniki pierwszej walidacji	51
5.3.2.	Wyniki drugiej walidacji	52
5.3.3.	Rozkłady błędów detekcji	53
5.3.4.	Analiza złożoności poszczególnych kroków metody	53
5.3.5.	Podsumowanie	54
6.	Podsumowanie	55

1. Wprowadzenie

W rozdziale tym zaprezentowano ogólne cele i wyzwania związane z tworzeniem automatycznych systemów wizyjnych oraz przedstawiono znaczenie detekcji obiektów w tychże systemach. Wymieniono także cele, dla jakich powstała niniejsza praca, a także streszczono jej zawartość na przestrzeni kolejnych rozdziałów.

1.1. Wykorzystanie komputerowych systemów wizyjnych

Dzięki coraz niższym cenom kamer cyfrowych wysokiej rozdzielczości, a także wyższej dostępności komputerów o dużej mocy obliczeniowej, temat budowy komputerowych systemów wizyjnych zyskał w ostatnich czasach dużą popularność. Ta rosnąca popularność z kolei, była motywacją do dużego, niespotykanego do tej pory, zainteresowania rozwojem w dziedzinie algorytmów związanych z przetwarzaniem i rozpoznawaniem obrazów. Efektem tego było powstanie, na przestrzeni ostatnich kilkunastu lat, w wielu ośrodkach badawczych na całym świecie, wielu propozycji metod do realizacji konkretnych zadań związanych z automatycznym rozpoznawaniem wizji, z których wiele z nich zostało użytych z powodzeniem w praktyce.

Wśród dziedzin, którym szybki rozwój systemów wizyjnych przynosi najwięcej korzyści można wymienić m.in.:

- informatykę
na systemach wizyjnych swą budowę opierają interfejsy multimodalne, rozpoznające np. gesty wykonywane przez operatora;
- automatykę
systemy wizyjne są podstawą działania robotów potrafiących podejmować decyzje na podstawie danych o świecie zewnętrznym;
- komunikację
dzięki systemom wizyjnym można pozyskiwać dane o natężeniu ruchu i na jego podstawie np. dynamicznie sterować sygnalizacją świetlną;
- przemysł samochodowy
systemy wizyjne wykorzystywane są w systemach zwiększających bezpieczeństwo jazdy, np. wykrywające znaki drogowe;

- systemy monitoringu

systemy wizyjne pozwalają na automatyczne wykrycie podejrzanych zachowań np. na lotnisku;

Głównym zadaniem komputerowego systemu wizyjnego jest wyodrębnienie z zadanego obrazu, statycznego bądź ruchomego, pewnej informacji a następnie wykonanie na jej podstawie pewnej czynności, najczęściej specyficznej dla konkretnego systemu. Zadanie to jest realizowane za pomocą kilku etapów, które mają na celu jak największe upodobnienie działania takiego systemu do ludzkiego widzenia:

- akwizycji obrazu;
- przetwarzania wstępnego obrazu;
- analizy obrazu;
- rozpoznania obrazu;
- podjęcia decyzji

W przypadku ludzkiego zmysłu wzroku, którego każdego dnia używamy wielokrotnie we wszystkich czynnościach życiowych, nie mamy nawet świadomości ich istnienia. Dzieje się tak dlatego, że wykonywane są one przez mózg w sposób bardzo szybki i naturalny. Jednak tworząc sztuczny system wizyjny musimy mieć świadomość złożoności procesu jaki prowadzi do podjęcia decyzji na podstawie obrazu. Dodatkowym jego utrudnieniem niech będzie fakt, że do tej pory sposób, w jaki ludzki mózg dokonuje analizy obrazu otrzymanego od narządu wzroku, nie został dokładnie zbadany. Oprócz tego, do niedoskonałości sztucznych systemów wizyjnych można zaliczyć m.in.

- niedoskonałość kamer, np. powstawanie szumów;
- utratę informacji w trakcie projekcji obrazu trójwymiarowego na dwuwymiarowy;
- zmiany oświetlenia obserwowanej sceny;
- częściowe lub całkowite zasłonięcie obiektów na scenie

Na dłuższą metę jednak, użycie komputerowych systemów wizyjnych niesie za sobą wiele korzyści w zastosowaniach, w których zostały one już dobrze sprawdzone i istnieje możliwość zastąpienia nimi ludzkiego zmysłu wzroku.

Należą do nich m.in:

- eliminacja błędów ludzkich wynikających z takich czynników jak np. stres;
- możliwość rozpoznawania obrazów poza zakresem światła widzialnego;
- możliwość umieszczenia systemu w miejscach niedostępnych lub nieoptymalnych dla człowieka;

- krótki czas reakcji

1.2. Znaczenie detekcji obiektów w komputerowych systemach wizyjnych

Jak wspomniano wyżej, w wielu swoich zastosowaniach systemy wizyjne opierają swoje działanie na detekcji pewnych obiektów na obrazie. Jest to jednak jeden z najbardziej wymagających problemów widzenia automatycznego, ze względu na to że na dwuwymiarowym rzucie obiektu na obraz, który dostępny jest do analizy, interesujący nas obiekt zmienia swój kształt w czasie, staje się on częściowo przysłonięty czy zmienia się poziom jego oświetlenia. W wielu przypadkach sama detekcja obiektu na obrazie jest tylko jednym z elementów jego działania odpowiedniego systemu wizyjnego. Na przykład w przypadku systemu, którego zadaniem jest analiza sposobu chodu człowieka, analiza taka dokonywana jest dopiero po wcześniejszym wykryciu obecności człowieka w widzialnej strefie przez odpowiedni system detekcji. Do innych zastosowań takich systemów zaliczają się np.: detekcja przechodniów w inteligentnych systemach prowadzenia pojazdów czy detekcja obiektów znajdujących się w zabronionej strefie (np. osoby niepożądanego na terenie prywatnym czy pozostawionej walizki na lotnisku).

Jak łatwo zauważyć, rozpoznanie konkretnych obiektów ma szerokie zastosowanie np. w systemach monitoringu, gdzie omyłkowe podniesienie alarmu w sytuacji potencjalnego zagrożenia może mieć znaczne konsekwencje. Dlatego też, niejednokrotnie, starając się opracować metodę wykrywającą pewną klasę obiektów na obrazie, poza osiągnięciem jak najlepszej trafności metody, szczególnie dużą uwagę poświęca się maksymalnie możliwemu wyeliminowaniu odsetku niepoprawnych wykryć zagrożenia, gdy takie nie istnieje.

Dodatkowym wyzwaniem dla autorów metod detekcji obiektów jest fakt, że nie istnieje metoda sprawdzająca się tak samo dobrze dla każdej klasy poszukiwanych obiektów. Wynika to z faktu istnienia różnych deskryptorów dla różnych klas obiektów, dzięki którym można w dobrym stopniu odróżnić obiekt należący do danej klasy od obiektu, który do niej nie należy (w przypadku detekcji człowieka może to być np. pewien sposób opisu sylwetki jego ciała, podczas gdy inny rodzaj obiektu będzie wykrywany w dużo lepszym stopniu wykorzystując informację o kolorze). Dlatego też w momencie budowy systemu wizyjnego przeznaczonego do detekcji konkretnej klasy obiektów, jego autorzy często zmuszeni są do przeprowadzenia szeregu badań w celu wybrania jak najlepszych deskryptorów dla danego problemu i wprowadzenia potencjalnych ulepszeń do istniejących już metod tak, aby w ich konkretnym zastosowaniu sprawdzały się jak najlepiej.

1.3. Rola deskryptorów cech w rozpoznawaniu obrazów

Obliczenie deskryptora cech polega na pewnym zmniejszeniu wymiarów potrzebnych do opisu pewnego zjawiska. Dobrze zbudowany deskryptor cech powinien działać w taki sposób, aby w procesie redukcji ilości informacji pomijać np. dane, których obecność powielona jest w kilku miejscach wejścio-

wej informacji, ale przede wszystkim pomijać informacje, które są bezużyteczne w kontekście danego problemu. Dobry deskryptor cech powinien także dodatkowo eksponować te informacje, które są najważniejsze dla danego problemu, tak aby, mając do dyspozycji tylko i wyłącznie wynik jego działania, można było w nieskomplikowany sposób stwierdzić jaka była oryginalna informacja, na podstawie której zbudowano wynik deskryptora.

Głównym powodem użycia deskryptorów cech jest zdecydowanie mniejsza złożoność czasowa i pamięciowa potrzebna do ich przetworzenia. Ma to szczególne znaczenie w przypadku zadania rozpoznawania obrazów, gdzie na wejściu zazwyczaj otrzymuje się obraz, często w wysokiej rozdzielczości. W takim wypadku analiza "piksel po pikselu" jest zbyt wymagająca czasowo, by można jej użyć w rzeczywistych systemach rozpoznawania obrazów. W celu poradzenia sobie z tym problemem, szuka się odpowiednich deskryptorów cech, które w dobry sposób utrzymują z wejściowego obrazu informacje ważne w kontekście zadania konkretnego systemu wizyjnego.

1.4. Cele pracy

Celem niniejszej pracy jest bezpośrednie porównanie wybranych deskryptorów cech używanych w metodach detekcji obiektów na obrazie cyfrowym. Zbadanie działania deskryptorów na dokładnie tym samym zestawie obrazów testowych i używając tego samego sposobu dokonywania klasyfikacji pozwoli w sposób wiarygodny porównać deskryptory pod względem jakościowym w zależności od charakterystyki klasy poszukiwanych obiektów, a także pod względem złożoności czasowej i pamięciowej. Dodatkowym celem pracy jest próba połączenia kilku z wybranych deskryptorów w jeden bardziej złożony i odpowiedź na pytanie czy wraz ze wzrostem pamięci i czasu potrzebnych do ich zbudowania można uzyskać rozsądną poprawę skuteczności detekcji obiektu konkretnej klasy na obrazie.

1.5. Zawartość pracy

W rozdziale 2 przedstawiono dotychczasową pracę naukową związaną z automatyczną detekcją obiektów na obrazie, ze szczególnym uwzględnieniem metod wykorzystujących deskryptory cech.

W rozdziale 3 zaprezentowano użyte w testach zestawy obrazów, metodologię przeprowadzonych testów, sposób implementacji środowiska testowego a także sposoby zbierania wyników do dalszej analizy.

W rozdziale 4 opisano w sposób szczegółowy zasady działania wybranych metod, sposób ich implementacji, a także zestawiono i przeanalizowano wyniki testów, jakim zostały poddane.

W rozdziale 5 zaprezentowano propozycje połączenia wybranych deskryptorów, analizując potencjalne korzyści jakie może nieść dane połączenie. Dodatkowo przeanalizowano i opisano wyniki testów, jakim zostały poddane kombinacje deskryptorów.

W rozdziale 6 podsumowano testy przeprowadzone w rozdziałach 4 i 5 i wyciągnięto wnioski na temat potencjalnego pola użycia przeanalizowanych metod.

2. Dotychczasowe badania

W rozdziale tym przedstawiono dotychczasowe badania nad wybranymi metodami detekcji obiektów na obrazie opierające się na deskryptorach cech, a także prace związane z ich łączeniem w celu poprawy skuteczności deskryptora.

2.1. Proste deskryptory

2.1.1. HOG

Metody Edge Orientation Histogram, zaprezentowana po raz pierwszy w pracy [FR94] oraz Scale-invariant feature transform, przedstawiona w pracy [Low04], stały się inspiracją dla stworzenia metody, którą zaproponowali Navneet Dalal i Bill Triggs w [DT05], której główną ideą jest budowa lokalnych histogramów orientacji krawędzi na przestrzeni całego obrazu. Autorom udało się zauważyć, że obecność obiektu i jego kształt można w dobry sposób opisać poprzez lokalny rozkład wartości gradientu i jego orientacji bez znajomości dokładnego położenia krawędzi.

Obraz wejściowy dzielony jest na bloki, które zostają z kolei podzielone na komórki. Deskryptor HOG wyliczany jest na podstawie długości i orientacji gradientu wyliczonego na obrazie wejściowym. W każdej komórce budowany jest histogram, w którym cechą jest kąt nachylenia gradientu, a liczebnością suma jego długości. W celu redukcji zjawiska aliasingu, krokiem poprzedzającym dodanie wartości do histogramu powinna być interpolacja, zarówno pomiędzy histogramami w sąsiadujących ze sobą komórkach jak i pomiędzy kątami. Kolejnym ważnym krokiem zaproponowanej metody jest normalizacja histogramów w obrębie bloków. Jest to spowodowane faktem, że w różnych częściach obrazu może się on różnić jasnością, co z kolei powoduje większy kontrast między tłem a obiektami pierwszego planu, co wpływa na wartość wyliczonego gradientu. Ostatecznie, deskryptor dla obrazu wejściowego reprezentowany jest przez połączenie histogramów na przestrzeni całego obrazu.

Aby jednak możliwe było wykrycie obiektu na realnym obrazie, np. pochodzącym z kamery, autorzy stosują metodologię tzw. ruchomego okna, polegającą na podzieleniu obrazu na fragmenty obrazu zgodne co do rozmiaru z obrazami treningowymi i dokonania próby detekcji na podstawie deskryptora wyliczonego właśnie w każdym z tych fragmentów.

Dodatkowo, zaproponowane podejście, polegające na niedokładnym odwzorowaniu przestrzeni w swoim deskrypcorze (użycie stosunkowo dużych bloków i komórek jako lokalnej reprezentacji obszaru obrazu) i stosunkowo dokładnego odwzorowania kąta nachylenia krawędzi, pozwala na wiarygodną reprezentację obiektów, które były przedmiotem badań autorów, czyli ludzi. Wynika to z faktu, że przemieszczenie

ciała na obrazie wpływa na zmianę wartości deskryptora w nieznacznym stopniu. Ponadto, aby możliwe było wykrycie obiektu, który na obrazie wejściowym ma inną skalę niż obiekt na obrazie treningowym, konieczne jest także jego kilkukrotne przeskalowanie i wynik każdego skalowania poddany detekcji tak, jakby był zupełnie niezależnym obrazem wejściowym.

Efektem badań była nowa metoda detekcji obiektów, która bardzo dobrze sprawdzała się w przypadku detekcji ludzi na obrazie i w momencie publikacji pozwalała na uzyskanie najlepszych wyników z dostępnych w tym czasie metod, w sytuacji gdy człowiek na obrazie jest wcale lub w niewielkim stopniu przysłonięty przez inne obiekty.

Poza nową metodą, dającą bardzo dobre wyniki w detekcji ludzi na obrazie, osiągnięciem związanym z pracą nad [DT05] jest budowa nowej bazy obrazów ludzi. Baza ta zawiera zbiór 1805 obrazów ludzi, na zróżnicowanym tle i w różnych pozycjach. Dodatkowo, zdjęcia te są niepozowane, co oznacza że ludzie na nich się znajdujący przyjmują szereg naturalnych pozycji ciała. Baza ta jest dostępna pod adresem <http://lear.inrialpes.fr/data>.

Pod względem zróżnicowania baza ta przewyższa dotychczas przygotowane i została później użyta w wielu badaniach dotyczących detekcji ludzi na obrazie.

2.1.2. Kowariancja cech

Deskryptor wykorzystujący do opisu cech inne podejście statystyczne niż histogram został zaprezentowany przez O. Tuzel i in. w pracy [TPM06], polegający na użyciu jako deskryptora cech macierzy kowariancji. Podobnie jednak jak w przypadku metody HOG do opisu pewnego obszaru obrazu używany jest histogram, tak w tej metodzie macierz kowariancji również używana jest jako lokalny deskryptor. Metoda ta z powodzeniem została użyta np. w detekcji ludzi na obrazie, czy do rozpoznania tekstur.

Założeniem metody jest użycie wielu informacji, jakie można wydobyć z obrazu, takich jak wartość piksela czy gradient, przy jednoczesnej redukcji ilości wymiarów jakie są potrzebne do ich zapisania. Metoda polega na zapisaniu dla każdego piksela obrazu wejściowego odpowiadającego mu zestawu kilku cech. Mogą to być np. m.in.: pozycja w poziomie, pozycja w pionie, jasność w skali szarości lub w składowych barwnych, wartość gradientu w każdym z kierunków i wypadkowy, kąt nachylenia gradientu. Tak jak w przypadku HOG, ponieważ macierz kowariancji opisuje lokalny fragment obrazu, konieczne jest podzielenie obrazu wejściowego na bloki o odpowiednim rozmiarze, a wynikowy deskryptor cech jest złożeniem macierzy kowariancji wyliczonych w obrębie wszystkich takich bloków.

Bardzo ważnym spostrzeżeniem zawartym w pracy [TPM06] jest użycie *integral images* do obliczenia kowariancji. Dzięki zastosowaniu tej metody, istnieje możliwość obliczenia sumy pikseli dowolnego prostokątnego obszaru obrazu w stałym czasie, co znacznie obniża złożoność implementacji całej metody.

Bardzo podobnie jak w przypadku powyższej metody, pojedynczej detekcji obiektu dokonuje się w zadanym ruchomym oknie, które przesuwane jest po całym, kilkukrotnie przeskalowanym, rzeczywistym obrazie wejściowym.

Efektem pracy [TPM06] było zaimplementowanie deskryptora cech, który może być obliczony dużo szybciej i wykrywać obiekty, w tym wypadku ludzi na obrazie, z lepszą skutecznością niż dotychczasowe

metody, w tym HOG. Wynika to z faktu, że metoda ta korzysta z większej liczby informacji zawartych na obrazie.

Ponieważ macierz kowariancji nie leży w przestrzeni euklidesowej, a w rozmaiłości Riemanna, przed użyciem klasyfikatorów działających w tej pierwszej konieczne jest rzutowanie deskryptora na przestrzeń euklidesowej poprzez operację logarytmowania.

2.1.3. LBP

Kolejną metodą wykorzystującą lokalne właściwości obrazu, jednak przy użyciu dużo prostszych założeń, jest metoda LBP (Local Binary Patterns - Lokalne Binarne Wzorce). Została ona po raz pierwszy zaprezentowana przez T. Oja i innych w pracy [OPH96], gdzie została użyta do klasyfikacji tekstur. W późniejszym czasie jednak, została ona użyta z powodzeniem także do innych celów związanych z detekcją obiektów, takich jak np. do rozpoznawania twarzy w pracy [AMH⁺06], czy do klasyfikacji mimiki twarzy w pracy [ZP07].

Deskryptorem w niniejszej metodzie jest histogram, wspólny dla całego opisywanego obrazu, w którym cechą jest liczba naturalna. Ideą metody jest poruszanie się elementem strukturalnym o zadanym rozmiarze $n \times n$ (zazwyczaj $n = 3$) po całym obrazie i porównywaniem wartości pikseli znajdujących się na jego obwodzie z wartością piksela w jego środku. W przypadku gdy różnica pomiędzy wartościami przekracza wartość zadanego progu w miejscu piksela na obwodzie wstawia się wartość 1, a w przeciwnym wypadku wartość 0. W ten sposób, z nowych wartości pikseli na obwodzie buduje się liczbę binarną długości $4(n - 1)$ bitów. Następnie, liczebność histogramu dla otrzymanej liczby jest zwiększana o 1.

Kwadratowa złożoność pamięciowa metody, a także możliwość pojawienia się sytuacji, w której dwa bliskoznaczne wzorce klasyfikowane są w odległych od siebie częściach histogramu, sprawia że oryginalna forma metody nie sprawdza się najlepiej w detekcji obiektów. Jej ulepszenie w tym kontekście zostało zaproponowane przez Y. Mu i in. w pracy [MYL⁺08], polegające na reinterpretacji histogramu. W ulepszonej metodzie wyniku operacji z elementem strukturalnym na obrazie nie interpretuje się jako liczby naturalnej, lecz jako łuk o pewnej długości, którego początek stanowi z początkiem układu współrzędnych pewien kąt. Deskryptorem z kolei jest dwuwymiarowy histogram, w którym jedną cechą stanowi długość łuku, a drugą kąt.

Podobnie jak w powyższych metodach, detekcji dokonuje się w ruchomym oknie, przesuwanym po kilkukrotnie przeskalowanym całym wejściowym obrazie.

Wnioskiem płynącym z pracy [MYL⁺08] jest opracowanie metody o niskiej złożoności pamięciowej, osiągającej wyniki lepsze od ww. metod, szczególnie w przypadkach, gdy obiekt jest częściowo zasłonięty. Wynika to z faktu, że metoda ta, w odróżnieniu do powyższych, nie opiera się bezpośrednio na obliczaniu gradientu, co może w pewnych przypadkach prowadzić do utraty informacji związanych z kształtem.

2.1.4. Edgelets

Nieco odmienne do zaprezentowanych do tej pory założenie przyjęli autorzy pracy [WN05] B. Wu i R. Nevatia, którzy zdecydowali się wykrywać obiekt, w tym wypadku ponownie człowieka, na

obrazie opierając się na informacjach dotyczących różnych części ciała. Takie podejście pozwala na poradzenie sobie z poprawną detekcją obiektu nawet w przypadku, gdy jest on częściowo przysłonięty przez inny a dodatkowo czyni to metodę bardziej odporną na zmianę pozycji, w jakiej znajduje się człowiek.

Autorzy do opisu sylwetki ludzkiej wprowadzili nowy rodzaj deskryptora, nazwany przez nich *edgelets*. Edgelet jest to krótki fragment linii lub krzywej. W początkowej fazie dokonuje się generacji wszystkich takich fragmentów, dzięki którym można opisać w mniej lub bardziej wiarygodny sposób interesującą nas klasę obiektu. W założeniach autorów, powinno być możliwe ułożenie uśrednionej ludzkiej sylwetki z dostępnych, wygenerowanych wcześniej *edgeletów*. Do detekcji używana jest funkcja opisująca podobieństwo pomiędzy wycinkiem obrazu, a kolejnymi *edgeletami*.

Metoda bazuje na obliczeniu podobieństwa pomiędzy gradientem obrazu wejściowego a fragmentami wzorcowymi, na zasadzie przesuwnego okna. Do obliczenia wartości podobieństwa konieczna jest znajomość wartości gradientu w danym punkcie obrazu, a także obliczenia iloczynu skalarnego między kątem nachylenia krawędzi na obrazie jak i w *edgelecie*. Jednak zastosowane uproszczenie, polegające na mocnej dyskretyzacji kątów nachylenia pozwala na znaczne zmniejszenie złożoności czasowej potrzebnej na obliczenie podobieństwa. Co ważne, w fazie treningowej, dokonuje się obliczenia podobieństwa między bazą *edgeletów* nie tylko na całej sylwetce ludzkiej, ale tak jak wspomniano także na jej częściach: torsie, nogach i głowie z ramionami.

Tak jak w dotychczas już przedstawionych metodach, próba detekcji dokonywana jest w ruchomym oknie, które przesuwane jest po kilkukrotnie przeskalowanym całym obrazie wejściowym.

Efekt badań nad pracą [WN05] był system pozwalający na wykrywanie ludzi na obrazie, bazujący na częściach ciała, a zatem pozwalający na wykrycie osób zasłoniętych przez inne obiekty nawet w dużym stopniu, w zadowalający sposób. Jednak jak przyznali autorzy, skala w jakiej zostały wygenerowane przez nich wzorcowe *edgelety* była przyczyną zbyt wysokiego odsetka nietrafionych detekcji.

2.2. Złożone deskryptory

Odmienne podejście do rozpoznania obiektu pomiędzy metodami pozwala na wykorzystanie zalet wszystkich metod przy jednoczesnej redukcji ich wad. Dzieje się to jednak kosztem wydłużonego wektora cech w nowym deskrytorze, co nie tylko wpływa na złożoność pamięciową samego deskryptora, ale także wydłuża czas zarówno treningu jak i klasyfikacji w nowo powstałej metodzie.

Przykładem użycia połączonych deskryptorów może być praca G. Gan i J. Cheng [GC11], w której to autorzy postanowili połączyć zaprezentowane wyżej metody HOG i LBP. Powodem wyboru tych deskryptorów były reprezentacja sylwetki przez kształt i orientację krawędzi w metodzie HOG i reprezentacja tekstur takich jak twarz czy ramiona przez metodę LBP. Połączenie tych dwóch podejść powinno zatem pozwolić na dużo lepszą jakość deskryptora niż w przypadku tych metod działających niezależnie od siebie.

Eksperymenty dowodzą, że istotnie nowy deskryptor cech powstały przez połączenie deskryptorów HOG i LBP ma wyższą skuteczność od obu metod przetestowanych niezależnie od siebie. Test przeprowadzono na przykładzie detekcji ludzi na obrazie, z wykorzystaniem wspomnianej już bazy INRIA.

2.3. Podsumowanie

Do detekcji obiektów danej klasy na obrazie cyfrowym wykorzystano do tej pory z powodzeniem szereg deskryptorów cech. Różnice między stosowanymi deskryptorami wynikają z informacji z obrazu wejściowego, jakie brane są pod uwagę przy tworzeniu deskryptora, regionu obrazu w jakim liczone są jednorazowo cechy i metody statystycznej służącej do opisu tych cech. Metody te różnią się zatem między sobą zarówno złożonością czasową i pamięciową. Dodatkowo, informacje z obrazu wejściowego, które brane są pod uwagę w trakcie tworzenia deskryptora mają wpływ na skuteczność detekcji obiektów - np. w jednym scenariuszu lepiej sprawdzać się będą informacje uzyskane na podstawie wyliczonego gradientu, a w drugim na podstawie informacji o jasności.

Ponadto, połączenie deskryptorów pozwalana na połączenie zalet kilku z nich, a co za tym idzie na poprawę skuteczności metody. Odbywa się to jednak kosztem zwiększonej złożoności pamięciowej i obliczeniowej.

3. Metodologia przeprowadzonych badań

W rozdziale tym zaprezentowano metodologię przeprowadzania badań metody detekcji obiektu pod względem zarówno jakościowym jak i czasowym.

3.1. Implementacja

Przedstawiona w niniejszym rozdziale metodologia wraz z metodami zaprezentowanymi w rozdziałach 4 i 5 zaimplementowana została w języku C++, przy użyciu biblioteki OpenCV2 [Bra00]. Biblioteka ta, poza tym że udostępnia możliwość wykonywania wielu operacji na obrazach z poziomu jej interfejsu, udostępnia także wiele metod uczenia maszynowego, w tym Support Vector Machine, wykorzystanej w tej pracy.

Implementacja została udostępniona na licencji GNU/GPL, na portalu GitHub, pod adresem <http://github.com/studzien/sauce>.

3.2. Wykorzystywany zestaw danych

Do badania działania metod detekcji obiektów wybrano bazę danych opracowaną we francuskim instytucie INRIA, która powstała w trakcie badań nad pracą [DT05], opisującą wówczas nowe podejście do detekcji ludzi na obrazie. Baza ta zawiera obrazy przeznaczone do uczenia i testowania metod detekcji ludzi na obrazie podzielone na zbiory próbek: **pozytywnych**, zawierających ponad 1500 obrazów zawierających sylwetkę człowieka wraz z ich lustrzanymi odbiciami oraz **negatywnych**, zawierających blisko 1700 obrazów niezawierających takiej sylwetki.

Baza ta powstała w wyniku wykorzystania już istniejących zbiorów obrazów, przypadkowych obrazów znalezionych w sieci a także z prywatnych zbiorów autorów, ze zdjęć wykonywanych w długim okresie czasu. W przypadku sylwetek ludzkich, wiele z nich zostało wyodrębnionych ze zdjęć, na których znalazły się one nieświadomie (byli to np. przechodnie w tle) i w związku z tym można mówić o braku przyjętej przez nich specyficznej pozy. Obrazy obecne w zbiorze pozytywnym, zostały znormalizowane i ujednolicone w taki sposób, że zawierają one ludzką sylwetkę w centrum obrazu o rozmiarze 64x128 pikseli (część zbioru zawiera dodatkowy margines 16 lub 3 pikseli dla każdej z krawędzi obrazu, w celu uniknięcia warunków brzegowych). Przykładowe obrazy z pozytywnego zbioru zaprezentowano na rysunku 3.1.



Rysunek 3.1: Przykłady pozytywnych próbek ze zbioru INRIA

Ponadto, wszystkie obrazy będące częścią pozytywnego zbioru próbek, zawierają dodatkowy opis lokalizacji osób na nim w formacie *Pascal Challenge*. Pozwala to na sprawdzenie poprawności działania finalnej metody. Autorzy zbioru nie zapewnili negatywnego zbioru danych, odpowiadającego pozytywnym próbkom o rozmiarze 64x128 pikseli, pozostawiając wybór okien tego rozmiaru z większych negatywnych obrazów autorom konkretnej implementacji. Propozycja wyboru takich negatywnych okien została zaprezentowana w 3.4.

3.3. Klasyfikator - Support Vector Machine

Zadanie klasyfikacji polega na podziale zbioru danych na dwa zestawy: treningowy i testowy. Każda próbka w zbiorze treningowym dostaje etykietę – jest próbką pozytywną, reprezentującą zajście jakiegoś zjawiska, lub negatywną – wtedy reprezentuje brak jego zajścia. Dla każdej próbki z obu zestawów wyliczana jest wartość deskryptora cech, po czym wraz z odpowiednią etykietą, pozytywną lub negatywną, przekazywana jest ona do klasyfikatora w celu wytrenowania go. Po wytrenowaniu klasyfikatora, przy jego użyciu możliwa jest klasyfikacja, na bardzo podobnej zasadzie do fazy treningowej. Klasyfikator na podstawie przekazanego wektora cech przewiduje do której z dwóch klas obiektów należy próbka – pozytywnej czy negatywnej.

Support Vector Machine (SVM) jest bardzo często używaną i dającą dobre wyniki techniką, statystycznego uczenia maszynowego. Metoda zaproponowana przez dwójkę C. Cortesa i V. Vapnika w pracy [CV95], z matematycznego punktu widzenia jest problemem optymalizacyjnym postaci:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i$$

przy ograniczeniach:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

gdzie zbiór treningowy to zbiór par $(x_i, y_i), i \in \{1, \dots, l\}$,

$x_i \in R^n$ jest wektorem cech dla danej próbki,

$y_i \in \{-1, 1\}$ jest identyfikatorem klasy i-tej próbki. Intuicyjna ilustracja metody, na przykładzie dwuwymiarowego wektora cech została zaprezentowana na rysunku 3.2.

Wektory cech odwzorowywane są na przestrzeń o większej liczbie wymiarów poprzez funkcję ϕ . Zadaniem SVM jest znalezienie takiego podziału hiperpłaszczyzny, jak najbardziej odległej od granic

umiejscowienia wektorów obu klas. Funkcja $K(\mathbf{x}_i, \mathbf{x}_j) = \xi(\mathbf{x}_i)^T \xi(\mathbf{x}_j)$ nazywana jest funkcją jądra SVM (*kernel*).

Najczęściej używanymi funkcjami są:

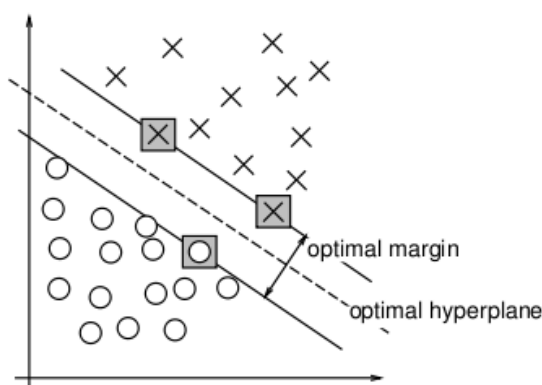
funkcja liniowa: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

funkcja wielomianowa: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d, \gamma > 0$

funkcja RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \gamma > 0$

esica (*sigmoid*): $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(-\gamma \mathbf{x}_i^T \mathbf{x}_j + r)$,

gdzie γ, r, d są parametrami.



Rysunek 3.2: Przykład budowy klasyfikatora dla wektora dwóch cech. Źródło: [CV95]

Zjawiskiem niepożądanym przy trenowaniu klasyfikatora jest jego przetrenowanie. Do takiej sytuacji dochodzi w momencie, gdy w zbiorze treningowym którejś z klas znajduje się zbyt duża liczba wektorów leżących bardzo blisko realnej granicy. Wtedy hiperpłaszczyzna wyznaczona przez SVM jest dobrze dopasowana do tych skrajnych w przypadku jednej klasy i próbek drugiej klasy, leżących dalej od realnej granicy. W konsekwencji, wyznaczona granica może leżeć dość daleko od realnej.

3.4. Podział zbioru danych

Jak wspomniano w 3.2, autorzy bazy danych INRIA zapewnili odpowiednio wykadrowany pozytywny zbiór treningowy i testowy, zawierający próbki o rozmiarze zgodnym z oknem detekcji, zawierające postać ludzką w samym jej centrum. Zbiory te zostały podzielone w stosunku 2416 próbek treningowych i 626 próbek testowych. W przypadku zbiorów negatywnych, które zostały podzielone w stosunku 1218 próbek treningowych i 453 próbki testowe, mamy do czynienia z pełnymi zdjęciami w różnych rozdzielczościach. W tym przypadku, autorzy zbioru pozostawili dowolność w konstrukcji wykadrowanych zbiorów autorom konkretnej implementacji.

Proponowane podejście do generacji zarówno zbioru treningowego i testowego jest następujące. Z każdego obrazu ze zbioru negatywnego generowane są cztery okna o rozmiarze 64x128 pikseli - trzy całkowicie losowe i jedno wybierane według następującej zasady: na całym obrazie stosowany jest operator Sobela w kierunkach poziomym i pionowym, wybierane jest to okno, dla którego suma wartości

bezwzględnych takiej operacji jest maksymalna. Innymi słowy, wybierane jest okno, które na całym obrazie dominuje pod względem ilości krawędzi.



Rysunek 3.3: Negatywny obraz ze zbioru danych i wygenerowane na jego podstawie negatywne dane treningowe, od lewej: na podstawie wyniku filtru Sobela i 3 losowe okna

Takie podejście powinno pozwolić na generację wiarygodnego, pierwszego klasyfikatora. Użyte negatywne dane, zarówno użyte do treningu klasyfikatora jak i jego walidacji, będą zróżnicowane pod względem ilości krawędzi, czyli cechy głównie wykorzystywanej przez prezentowane deskryptory do opisu obiektów.

Przykład negatywnego obrazu ze zbioru wraz z wygenerowanymi na jego podstawie oknami został przedstawiony na rys. 3.3.

3.5. Trening i walidacja

Wytrenowanie klasyfikatora Support Vector Machine polega na konstrukcji wektora próbki - w kontekście tej pracy jest to wynik obliczenia danego deskryptora cech lub wynik połączenia kilku deskryptorów. Jeżeli wyjściem z deskryptora jest macierz, dokonuje się rozwinięcia tej macierzy w wektor poprzez połączenie ze sobą kolejnych wierszy lub kolumn. Przyjęty sposób zmiany rozmiaru macierzy nie ma znaczenia, gdyż każdy element wektora reprezentuje niezależną od innych cechę, ważne jest natomiast by zachować konsekwencję zarówno w trakcie budowy takiego wektora w fazie treningu jak i w fazie klasyfikacji. Każdemu wektorowi próbki przypisana jest etykieta oznaczająca klasę, do jakiej on należy. W pracy przyjęto następujący sposób etykietowania: do klasy 0 należą wszystkie wektory, które są efektem zastosowania deskryptora na próbce nie zawierającej sylwetki ludzkiej, z kolei do klasy 1 na próbce z sylwetką ludzką.

W pracy do budowy klasyfikatora użyto implementacji Support Vector Machine udostępnionej w bibliotece OpenCV2. W trakcie badań uwzględniono budowę klasyfikatora z użyciem liniowego odwzorowania (tzw. *kernel*) wektora próbki.

Klasyfikator nie posiada, w przeciwieństwie do danych ze zbioru treningowego, żadnych informacji na temat klasy, do jakiej należą poddawane przez nas klasyfikacji próbki ze zbioru testowego, dlatego też pierwszym sprawdzeniem poprawności działania klasyfikatora powinno być zastosowanie go na przygotowanym wcześniej zbiorze testowym. Znajomość procentowej poprawności odpowiedzi klasyfikatora pozwala krytycznie spojrzeć na poprawność zastosowanej metody i zauważyć pewne problemy jak np.

przetrenowanie klasyfikatora, zanim dokona się sprawdzenia działania metody na rzeczywistych, bardziej złożonych danych.

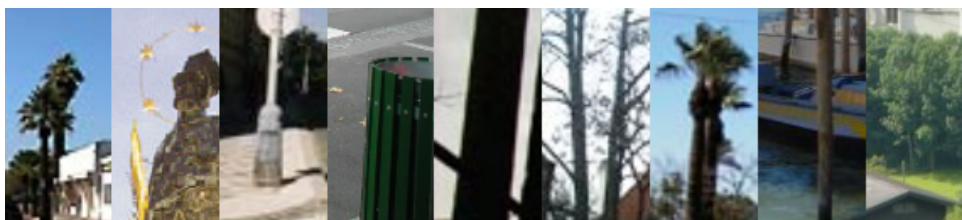
W przypadku detekcji obiektów, takich jak ludzie, na obrazie, za zadowalający można przyjąć klasyfikator zapewniający ok. 90% poprawności klasyfikacji zarówno na pozytywnym jak i negatywnym zbiorze testowym.

W przypadku Support Vector Machine, walidacja jest szczególnie ważna w przypadkach, w których używa się funkcji odwzorowującej wektor na wyższy wymiar. Wynika to z faktu, że dobór parametrów funkcji mapującej dokonywany jest na podstawie wyniku skuteczności klasyfikatora na zbiorze testowym.

3.6. Dotrenowanie klasyfikatora

Skorzystanie z samego, wcześniej przygotowanego zbioru treningowego, może nie dawać zadowalających rezultatów w przypadku dokonywania klasyfikacji na rzeczywistych obrazach. Najczęstszą tego przyczyną jest fałszywa pozytywna detekcja w oknach, w których obserwowalne są kształty w pewnym sensie zbliżone do ludzkich, np. zawierające długie pionowe równoległe linie. Przykłady kilku takich detekcji przedstawiono na rysunku 3.4.

W celu poprawienia skuteczności działania klasyfikatora dokonuje się jego dotrenowania. Polega ono na wykryciu fałszywych pozytywnych wskazań na testowym zbiorze obrazów rzeczywistych, a następnie ponownym wytrenowaniu klasyfikatora, tym razem jednak ze zbiorem negatywnych próbek powiększonym o właśnie te niepoprawne detekcje.



Rysunek 3.4: Przykłady fałszywie pozytywnych detekcji

Negatywny zbiór treningowy jest jednak powiększany nie przez całość, a tylko część fałszywych pozytywnych okien wygenerowanych w fazie testu. Badania pokazały, że zbyt duży udział takich próbek w całym negatywnym zbiorze treningowym prowadzi do zjawiska przetrenowania klasyfikatora, opisanego w 3.3. W związku z tym pierwotny zbiór treningowy powiększany jest np. o 1/3 losowych, wygenerowanych detekcji tego typu, po czym następuje ponowny trening klasyfikatora.

3.7. Detekcja w przesuwym oknie

Ze względu na fakt, że rzeczywiste obrazy, na których dokonuje się detekcji obiektów, mogą mieć różne rozdzielczości, krokiem wymaganym na samym początku procesu detekcji, po wczytaniu obrazu powinna być pewna jego normalizacja. W niniejszej pracy, wszystkie obrazy wejściowe zmniejszane

są do wysokości 600 pikseli, z zachowaniem proporcji. Dopiero na takim obrazie rozpoczynana jest detekcja obiektów.

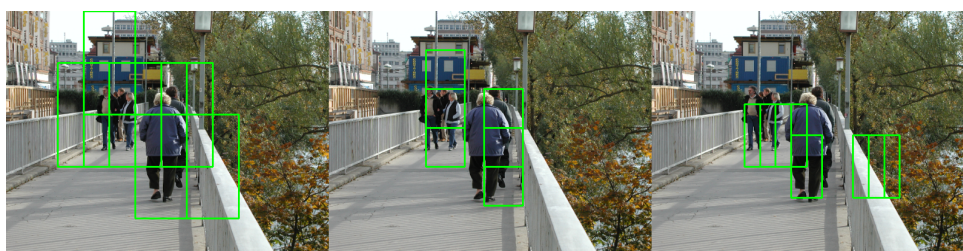
Biorąc pod uwagę wymagania klasyfikatora, polegające na tym że klasyfikacji można poddać tylko i wyłącznie okno o takim samym rozmiarze jak próbki poddane treningowi, niemożliwa jest klasyfikacja całego obrazu wejściowego w jednym kroku. Aby dokonać detekcji stosuje się koncepcję tzw. przesuwanego okna. Polega ona na podziale obrazu wejściowego na fragmenty o rozmiarze zgodnym z wielkością okna klasyfikowanego. Dla każdego takiego fragmentu dokonuje się osobnej, niezależnej od innych fragmentów, klasyfikacji. Takie niezłożone podejście ma jednak swoją wadę - ze względu na dużą ilość testowanych okien w obrębie jednego obrazu nawet wysoka celność klasyfikatora w przypadku próbek negatywnych może prowadzić do dużej ilości fałszywie pozytywnych detekcji.

W badaniach przeprowadzonych w niniejszej pracy dokonano detekcji w oknie o rozmiarze 64x128 pikseli. Okno było przesuwane co 32 piksele w poziomie i co 64 piksele w pionie.

3.8. Skalowanie obrazu testowanego

Ze względu na fakt, iż rzeczywiste obrazy testowane są zwykłymi fotografiami, zrobionymi w różnych okolicznościach, sylwetki ludzkie na nich obecne nie podlegają żadnej normalizacji. Oznacza to, że człowiek, który powinien zostać wykryty na obrazie, może zajmować różną jego część, w zależności od tego jak daleko od miejsca zrobienia zdjęcia znajdował się w chwili jego zrobienia. W testowym zbiorze danych obecne są przykłady zdjęć, w których postać znajduje się na pierwszym planie i jest obecna na prawie całej wysokości obrazu. Znajduje się w nim również wiele obrazów, w których postaci znajdują się w tle i wypełniają stosunkowo małą powierzchnię obrazu.

Przy stałej wielkości okna detekcji, która określona jest przez sposób budowy deskryptora cech w trakcie treningu klasyfikatora, w celu wykrycia obiektów o różnej wielkości na obrazie, konieczne jest zatem kilkukrotne przeskalowanie obrazu testowanego.



Rysunek 3.5: Przykłady detekcji na obrazie testowanych w skalach kolejno: **0.75**, **1** i **1.25**

Testy wykonane na obrazach przeskalowanych w skalach 0.25, 0.5, 0.75, 1 i 1.25 wykazały, że ostatecznie wystarczające będzie przeskalowanie obrazów testowanych w skalach **0.5**, **0.75** i **1**. W przypadku skali 0.25, pożądany obiekt był wykrywany w bardzo niewielkiej ilości przypadków, z kolei dla skali 1.25, ekstrakcja okien detekcji z dużym zbliżeniem, powodowała klasyfikację próbek z wieloma szczegółami obecnymi w otoczeniu. Konsekwencją tego była generacja znacznej ilości fałszywych pozytywnych detekcji.

Na rysunku 3.5 znajdują się przykłady detekcji obiektów na przeskalowanych obrazach testowych w skalach 0.75, 1 i 1.25.

3.9. Eliminacja redundantnych detekcji

Detekcja obiektu na kilku przeskalowanych obrazach a także mały krok przesunięcia przesuwającego okna detekcji może prowadzić do sytuacji, w której ten sam obiekt zostanie wykryty kilkakrotnie. Aby w ostatecznym wyniku detekcji uniknąć tego typu sytuacji przyjęto następujący algorytm postępowania:

Dane: zbiór okien detekcji zaklasyfikowanych pozytywnie przez klasyfikator w postaci
 {współrzędne okna, pewność detekcji}

Wynik: zbiór ostatecznych okien detekcji

strefa zabroniona := \emptyset ;

finalny zbiór okien detekcji := \emptyset ;

posortuj wejściowe okna detekcji malejąco względem pewności detekcji;

foreach okno **do**

if okno = {współrzędne, pewność} \notin strefa zabroniona **then**

 finalny zbiór okien detekcji := finalny zbiór okien detekcji \cup okno;

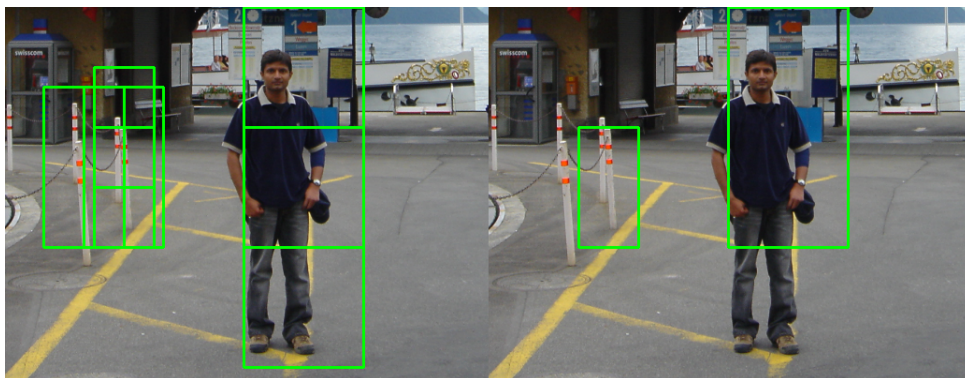
 strefa zabroniona := strefa zabroniona \cup współrzędne;

end

end

return finalny zbiór okien detekcji

Algorytm 1: Algorytm usuwania redundantnych detekcji



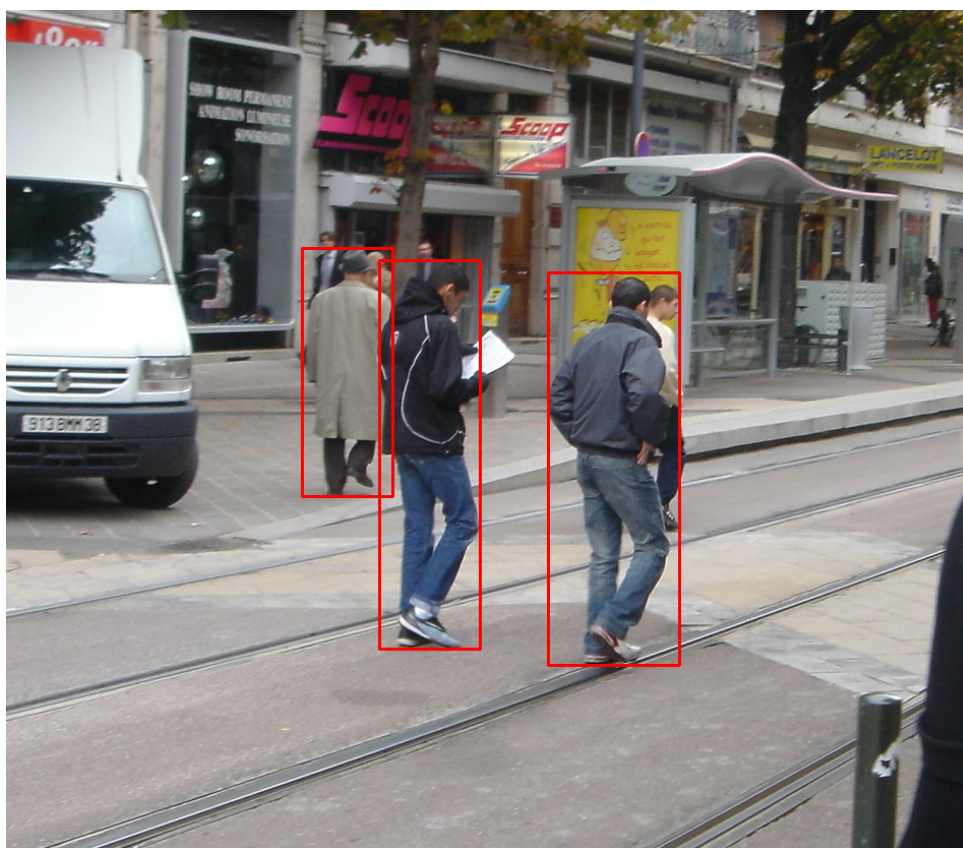
Rysunek 3.6: Po lewej obraz z redundantnymi detekcjami, po prawej po usunięciu redundantnych detekcji

Przykład pierwotnych, redundantnych okien detekcji wraz z oknami po redukcji, znajduje się na rysunku 3.6.

3.10. Walidacja z wykorzystaniem danych rzeczywistych

Dzięki udostępnieniu wraz z pozytywnymi zestawami (zawierającymi przynajmniej jedną sylwetkę ludzką), zarówno zdjęć testowych jak i treningowych w bazie INRIA plików opisu umiejscowienia tychże na obrazie w formacie *Pascal Challenge*, możliwe było zbudowanie automatycznego narzędzia pozwalającego na przebadanie skuteczności działania metody na obrazach jakie w rzeczywistości zostaną poddane procesowi detekcji.

Przykładowy obraz z pozytywnego zbioru testowego z zaznaczonymi występującymi na nim sylwetkami ludzkimi znajduje się na rysunku 3.7, z kolei odpowiadający mu plik opisującym na nim obecność sylwetek ludzkich w formacie *Pascal Challenge* znajduje się poniżej (oryginalny obraz ma rozmiar 953 x 828 pikseli).



Rysunek 3.7: Przykładowy obraz z pozytywnego zbioru testowego z zaznaczonymi występującymi na nim postaciami na podstawie informacji zawartych w odpowiadającym mu pliku w formacie *Pascal Challenge*

PASCAL Annotation Version 1.00

Image filename : "Test/pos/crop001514.png"

Image size (X x Y x C) : 953 x 828 x 3

Database : "The INRIA Rhone-Alpes Annotated Person Database"

Objects with ground truth : 3 { "PASperson" "PASperson" "PASperson" }


```
# Note that there might be other objects in the image
# for which ground truth data has not been provided.

# Top left pixel co-ordinates : (0, 0)

# Details for object 1 ("PASperson")
# Center point — not available in other PASCAL databases — refers
# to person head center
Original label for object 1 "PASperson" : "UprightPerson"
Center point on object 1 "PASperson" (X, Y) : (436, 280)
Bounding box for object 1 "PASperson" (Xmin, Ymin) – (Xmax, Ymax) : (367, 246)

# Details for object 2 ("PASperson")
# Center point — not available in other PASCAL databases — refers
# to person head center
Original label for object 2 "PASperson" : "UprightPerson"
Center point on object 2 "PASperson" (X, Y) : (345, 263)
Bounding box for object 2 "PASperson" (Xmin, Ymin) – (Xmax, Ymax) : (291, 234)

# Details for object 3 ("PASperson")
# Center point — not available in other PASCAL databases — refers
# to person head center
Original label for object 3 "PASperson" : "UprightPerson"
Center point on object 3 "PASperson" (X, Y) : (609, 293)
Bounding box for object 3 "PASperson" (Xmin, Ymin) – (Xmax, Ymax) : (533, 258)
```

Jak wspomniano w 3.7, nawet w rzeczywistych systemach wizyjnych, których zadaniem jest detekcja obiektów na obrazie, do detekcji używa się przesuwanego okna przesuwanego wzdłuż i w szerz całego testowanego obrazu, a klasyfikator klasyfikuje każde z nich jako pozytywne lub negatywne. W celu stwierdzenia, czy rezultat dla okna uzyskany w wyniku klasyfikacji jest poprawny w kontekście informacji zawartych w pliku opisującym informacje o sylwetkach obecnych na obrazie, przyjęto następującą procedurę postępowania:

Dane: współrzędne okna detekcji, współrzędne prostokątów opisujących obiekty obecne na obrazie

Wynik: wartość logiczna odpowiadająca na pytanie, czy okno jest pozytywne czy negatywne

foreach *prostokąt opisujący obiekt* **do**

if *okno ma przynajmniej jeden punkt wspólny z prostokątem* **then**

if $\text{pole}(\text{okno} \cap \text{prostokąt}) \geq \frac{1}{2} \min(\text{pole}(\text{okno}), \text{pole}(\text{prostokąt}))$ **then**

return *true*;

end

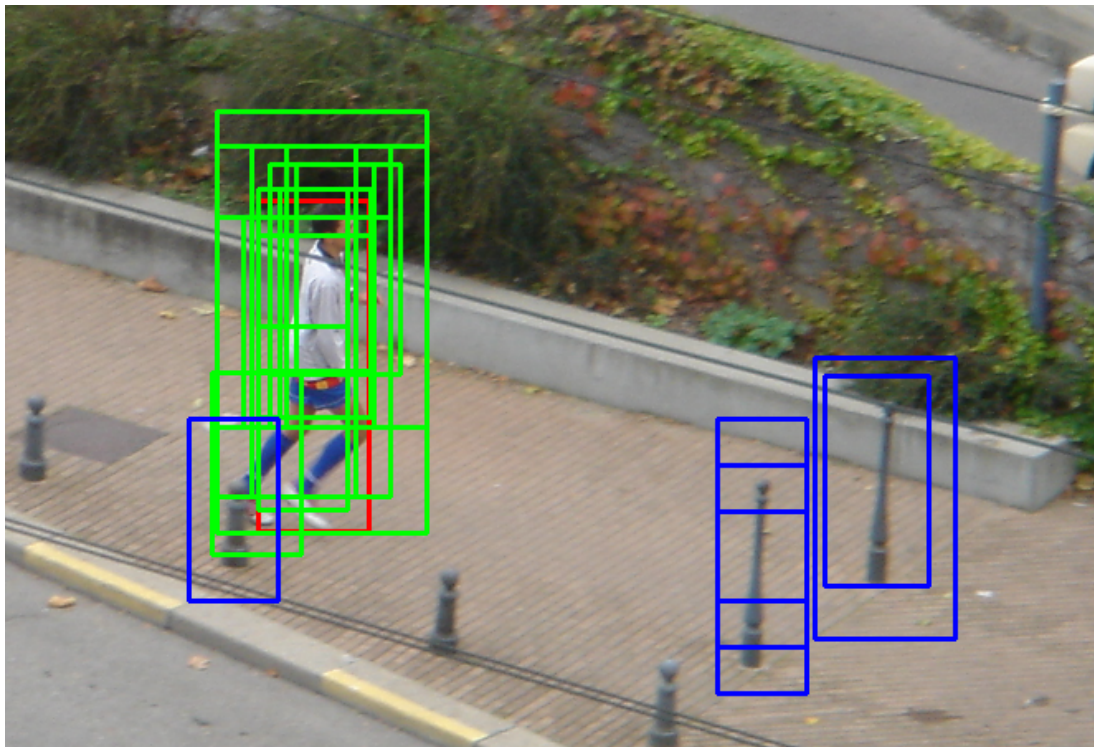
end

end

return *false*

Algorytm 2: Algorytm sprawdzenia czy okno poddane detekcji jest w rzeczywistości pozytywne czy negatywne

Na rysunku 3.8 zaprezentowano przykład działania powyższego algorytmu, na przykładzie detekcji wykonywanej na kilkakrotnie przeskalowanym obrazie. Kolorem czerwonym zaznaczony jest prostokąt opisujący obiekt na obrazie, na podstawie załączonego pliku. Kolorem niebieskim oznaczone są okna zaklasyfikowane jako fałszywie pozytywne detekcje. Kolorem zielonym oznaczone są okna zaklasyfikowane jako prawdziwie pozytywne detekcje.



Rysunek 3.8: Przykłady klasyfikacji detekcji okien.

3.11. Sposób bezpośredniego porównania metod

Porównania skuteczności metod detekcji obiektów można dokonać poprzez sprawdzenie jej działania na bardzo zróżnicowanym zbiorze testowym. W kontekście tej pracy jest to zbiór różnych zdjęć, na których obecni są ludzie w różnych pozach, w różnych częściach zdjęcia, a także na bardzo zróżnicowanym tle. Zastosowanie detekcji przy użyciu każdej z poszczególnych metod pozwala na znalezienie zależności pomiędzy odsetkiem fałszywych pozytywnych detekcji a skutecznością detekcji. Sprawdzenie metody pod takim kątem pozwala na znalezienie konsensusu pomiędzy stopniem fałszywych pozytywnych detekcji jakie jesteśmy w stanie zaakceptować w konkretnym zastosowaniu a celnością detekcji, jaką jesteśmy w stanie osiągnąć przy poszukiwaniu na obrazie konkretnej klasy obiektów. W rzeczywistych przypadkach bowiem, z reguły nie mamy do czynienia ze wspomnianym przypadkiem detekcji na bardzo zróżnicowanym zbiorze, lecz w stosunkowo niezmiennym środowisku, które znane jest z góry. Wyznaczenie zależności pomiędzy częstością występowania fałszywych pozytywnych detekcji w przypadku konkretnego obrazu a skutecznością metody pozwala zatem odpowiedź na pytanie - do jakiego poziomu należałoby wytrenować klasyfikator, aby można było uzyskać zadowalający oczekiwany stopień detekcji danej klasy obiektów. Bardzo pomocnym narzędziem w tego typu analizie jest wyznaczenie rozkładu błędu detekcji (DET - Detection error tradeoff), zaproponowanego w [MDK⁺97]. Wykres, który służy do porównania między sobą klasyfikatorów binarnych, przedstawia zależność między odsetkiem fałszywych pozytywnych detekcji (reprezentowanej na osi odciętych wykresu), a stopniem nietrafionych pozytywnych detekcji danej metody (reprezentowanej na osi rzędnych wykresu). W przypadku dokonywania detekcji w przesuwym oknie, jako odsetek fałszywych pozytywnych detekcji przyjmuje się rząd stosunku fałszywie pozytywnie zaklasyfikowanych okien do wszystkich badanych negatywnych okien (FPPW - False Positives Per Window). Wyższość jednej metody nad drugą, na podstawie wykresu DET, stwierdza się wtedy, gdy dla tych samych wartości fałszywie pozytywnych wskazań, wykres dla metody przyjmuje mniejszą wartość nietrafionych detekcji (krzywa opisująca rozkład lepszej metody jest na wykresie "pod"wykresem metody uznawanej za gorszą).

Oprócz tego, metody można bezpośrednio porównać wtedy i tylko wtedy jeżeli dokona się detekcji w przesuwym oknie z takim samym przesunięciem okna w pionie i w poziomie w kolejnych detekcjach. Ponieważ chcemy bardzo dokładnie przebadать zależność pomiędzy ilością fałszywych wskazań a skutecznością detekcji ilość okien, w jakich dokonywana jest detekcja przy użyciu każdej z metod powinna być taka sama i możliwie jak największa.

3.12. Analiza złożoności metod

Zaproponowane metody różnią się między sobą m.in. operacjami matematycznymi potrzebnymi do obliczenia danego deskryptora czy długością wektora wynikowego. Dlatego też, poza przeprowadzoną analizą jakościową wybranych metod, przeprowadzono analizę czasową badanych metod. Badając każdą z nich zmierzono czas, jaki był potrzebny dla każdego z ww. etapów przeprowadzonych badań. Aby zapewnić wiarygodną porównywalność uzyskanych wyników wszystkie badania przeprowadzono na identycznej konfiguracji sprzętowej, a proces zajmujący się obliczeniami opisanymi w tym rozdziale był

jedynym aktywnie działającym w systemie, przez co nie był on wywłaszczany przez inne procesy w kontekście czasu zajętości procesora.

Konfiguracja testowa była następująca:

- CPU: 4 x 2.40 GHz
- RAM: 14 GB
- System operacyjny: Amazon Linux AMI 2012.09

W tym miejscu należy dodać, że sposób implementacji metod nie zapewniał zrównoleglenia ich obliczeń na wielu rdzeniach procesora. Toteż w obliczeniach, przy powyższej konfiguracji, w jednym czasie, aktywnie był wykorzystywany tylko i wyłącznie jeden rdzeń procesora.

3.13. Podsumowanie

W celu podsumowania obranej metodologii przeprowadzonych badań, w poniższej tabeli zestawiono procedurę "krok po kroku" przyjętą w metodologii analizy wybranych deskryptorów używanych w algorytmach detekcji obiektów. Oprócz nazwanych etapów prowadzących do uzyskania oceny danej metody uwzględniono informacje wejściowe potrzebne do przeprowadzenia każdego z kolejnych kroków jak i informacje dostępne na wyjściu, będące efektem przeprowadzenia danego etapu.

Lp.	Nazwa kroku	Wejście	Wyjście
1.	Generacja negatywnego zbioru testowego	Zbiór negatywnych 1221 zdjęć (niezawierających sylwetek ludzkich)	Negatywne kadry treningowe 64x128 pikseli - kadr o największej ilości krawędzi dla danego zdjęcia oraz 5 losowych kadrów
2.	Ekstrakcja cech z pozytywnego zbioru treningowego	Pozytywny zbiór kadrów treningowych o rozmiarze 64x128 pikseli 2419	Macierz zawierająca 2419 wierszy, w każdym zawierająca wektor deskryptora cech
3.	Ekstrakcja cech z negatywnego zbioru treningowego	Negatywny zbiór kadrów treningowych (wyjście z 1) o rozmiarze 64x128 pikseli	Macierz zawierająca 1221 wierszy, w każdym zawierająca wektor deskryptora cech
4.	Pierwszy trening klasyfikatora SVM	Macierz cech pozytywnych (wyjście z 2) jako dane klasy 1, macierz cech negatywnych (wyjście z 3) jako dane klasy 0	Struktura SVM
5.	Walidacja krzyżowa pierwszego klasyfikatora	Struktura wytrenowanego klasyfikatora SVM (wyjście z 4), zbiór pozytywnych (629) i negatywnych (909) kadrów testowych o rozmiarze 64x128	Skuteczność działania metody na pozytywnym i negatywnym zbiorze testowym

6.	Znalezienie fałszywych pozytywnych wskazań dla metody	Zbiór negatywnych (456) zdjęć	Negatywne kadry treningowe 64x128, które pierwszy klasyfikator zaklasyfikował jako próbkę pozytywną
7.	Ekstrakcja cech ze zbioru fałszywych pozytywnych wskazań	Kadry fałszywych pozytywnych wskazań dla metody	Macierz zawierająca n wierszy, w każdym zawierająca wektor deskryptora cech
8.	Połączenie deskryptorów negatywnych cech	Macierz deskryptora negatywnego zbioru treningowego (wyjście z 3, zawierająca 1221 wierszy), macierz deskryptora fałszywych pozytywnych wskazań (wyjście z 7, zawierająca m wierszy)	Wspólna macierz zawierająca 1221+m wierszy, w każdym zawierająca wektor deskryptora cech
9.	Trening klasyfikatora SVM	Macierz cech pozytywnych (wyjście z 2) jako dane klasy 1, macierz cech negatywnych (wyjście z 8) jako dane treningowe klasy 0	Dotrenowana struktura SVM
10.	Walidacja krzyżowa dotrenowanego klasyfikatora	Struktura dotrenowanego klasyfikatora SVM, zbiór pozytywnych i negatywnych (629 + 909) kadrów testowych o rozmiarze 64x128	Skuteczność działania metody na pozytywnym i negatywnym zbiorze testowym po dotrenowaniu
11.	Detekcja sylwetek ludzkich na obrazach testowych	Zestaw (629) obrazów testowych z sylwetkami ludzkimi wraz z opisem ich umiejscowienia	Rozkład DET (ilość fałszywych pozytywnych detekcji do stopnia celności) dla metody, wyniki detekcji dla poszczególnych obrazów testowych

Tablica 3.1: Podsumowanie metodologii przyjętej w trakcie badania wybranych metod

4. Wybrane deskryptory cech

4.1. HOG - Histogram of Oriented Gradients

4.1.1. Algorytm obliczenia deskryptora

Algorytm obliczenia deskryptora cech HOG ma następującą postać:

Dane: obraz $n \times m$ w przestrzeni barw RGB, k - rozmiar komórki, w - szerokość histogramu (tutaj 9)

Wynik: wektor deskryptora cech o długości $\lfloor \frac{m}{k} \rfloor \times \lfloor \frac{n}{k} \rfloor \times w$

zamień przestrzeń barw RGB na skalę szarości;

oblicz gradient w obu kierunkach na obrazie np. używając maski $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$;

dla każdego piksela obrazu oblicz długość wektora gradientu ($\sqrt{I_x^2 + I_y^2}$) oraz jego kąt, tak aby leżał w przedziale $[0; \pi]$ (wykorzystując funkcję $\text{atan2}(I_y, I_x)$);

podziel obraz na komórki $k \times k$;

dla każdej komórki utwórz histogram o szerokości w , gdzie cechą jest kąt z przedziału $[0; \pi]$, a liczebnością suma długości gradientów;

foreach $\text{piksel} \in \text{obraz}$ **do**

 dokonaj interpolacji dwuliniowej współrzędnych piksela względem środków czterech najbliższych komórek;

 w histogramach odpowiadającym powyższym komórkom zwiększ liczebność kątom, po przeprowadzeniu interpolacji liniowej kąta odpowiadającemu pikselowi;

end

dokonaj normalizacji każdej próbki (w niniejszej pracy jest to norma L2);

połącz kolejne wiersze macierzy próbek w wektor;

return *wektor cech*

Algorytm 3: Algorytm obliczenia deskryptora cech HOG

4.1.2. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji pierwszego klasyfikatora wytrenowanego próbkami, uzyskanymi metodą HOG, w zależności od rozmiaru komórki, w której wyliczany był histogram orientacji gradientów.

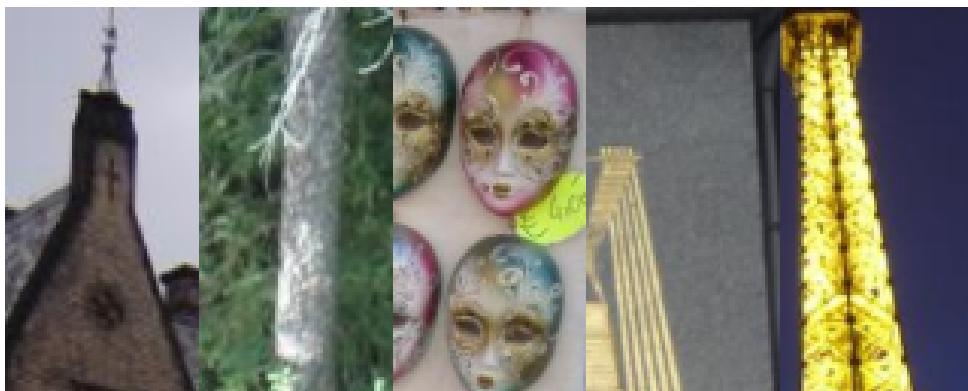
Rozmiar komórki	8x8	12x12	16x16
-----------------	-----	-------	-------

Funkcja jądra SVM	liniowa	liniowa	liniowa
Długość wektora cech	7308	800	512
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	575	474	462
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	57	152	164
Celność metody w pozytywnym zbiorze testowym	91,85%	75,72%	73,81%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	31	172	57
Liczba negatywnych detekcji w negatywnym zbiorze testowym	875	734	849
Celność metody w negatywnym zbiorze testowym	96,58%	81,02%	93,71%

Tablica 4.1: Wyniki pierwszej walidacji krzyżowej przy użyciu deskryptora HOG

4.1.3. Przykłady fałszywych detekcji

Przeważającą część fałszywie pozytywnych detekcji uzyskiwanych metodą HOG są okna, w których występują pary długich, równoległych linii, przypominające ludzką sylwetkę czy elementy które kształtem przypominają ludzką twarz. Jest to spodziewany efekt, biorąc pod uwagę to jak budowany jest deskryptor cech (opisywany jest kształt na podstawie krawędzi obrazu wejściowego). Ilości wygenerowanych fałszywie pozytywnych okien, w zależności od rozmiaru komórki, przedstawione zostały w tabeli 4.1.3. Przykładowe fałszywie pozytywne detekcje dla metody HOG zostały przedstawione na rysunku 4.1.



Rysunek 4.1: Przykłady fałszywie pozytywnych detekcji w metodzie HOG

Rozmiar komórki	8x8	12x12	16x16
Funkcja jądra SVM	liniowa	liniowa	liniowa
Ilość znalezionych pozytywnie fałszywych okien	2759	14538	6446

Tablica 4.2: Ilość znalezionych fałszywie pozytywnych okien przy użyciu deskryptora HOG

4.1.4. Wyniki drugiej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji klasyfikatora po dotrenowaniu pierwszego klasyfikatora fałszywie pozytywnymi detekcjami, wykrytymi w zbiorze testowym. W celu zminimalizowania efektu przetrenowania klasyfikatora wzięto pod uwagę tylko 25% wygenerowanego zbioru.

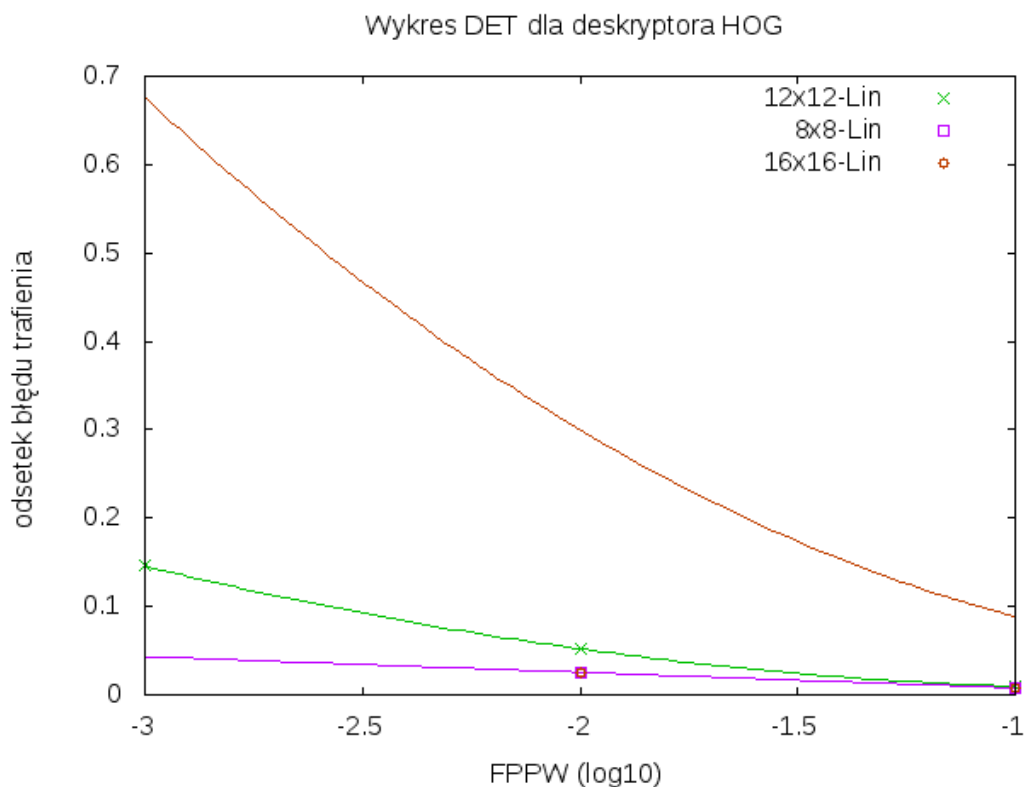
Rozmiar komórki	8x8	12x12	16x16
Funkcja jądra SVM	liniowa	liniowa	liniowa
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	531	445	223
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	95	181	403
Celność metody w pozytywnym zbiorze testowym	84,82%	71,09%	35,62%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	32	79	290
Liczba negatywnych detekcji w negatywnym zbiorze testowym	874	827	614
Celność metody w negatywnym zbiorze testowym	96,47%	91,28%	67,77%

Tablica 4.3: Wyniki drugiej walidacji krzyżowej przy użyciu deskryptora HOG

4.1.5. Rozkłady błędów detekcji

Wykres rozkładu błędu detekcji dla metody HOG zaprezentowany został na rysunku 4.2. Na osi odciętych wykresu znajduje się rząd stosunku fałszywych pozytywnych detekcji klasyfikatora i wszystkich negatywnych testowanych okien, na osi rzędnych zaś, średni odsetek błędu klasyfikatora w detekcji

pozytywnego okna. Na podstawie wykresu można wyciągnąć wniosek, że w przypadku użycia liniowej funkcji jądra klasyfikatora SVM, najlepsze zachowanie prezentuje klasyfikator zbudowany przy deskrytorze HOG wyliczanego na komórkach o rozmiarze 8x8 pikseli. Zachowuje on bardzo dobry wynik detekcji obiektu (ok. 95%), w przypadku, gdy ilość FPPW jest rzędu 10^{-2} . Na podstawie wykresu można również stwierdzić, że przy wyliczaniu bloków o rozmiarze 16x16 pikseli deskryptor HOG jest złym deskrytorem, a stopień jego błędu detekcji rośnie aż do 70% przy FPPW rzędu 10^{-3} .



Rysunek 4.2: Rozkład błędu detekcji dla różnych przyjętych komórek

4.1.6. Analiza złożoności poszczególnych kroków metody

W poniższej tabeli zestawiono czasy wykonywania poszczególnych kroków metodologii, w zależności od rozmiaru komórki deskryptora HOG, na konfiguracji testowej.

Rozmiar komórki	8x8	12x12	16x16
Funkcja jądra SVM	liniowa	liniowa	liniowa
Ekstrakcja cech pozytywnych / próbka	6,23 ms	5,67 ms	5,19 ms
Ekstrakcja cech negatywnych / próbka	5,27 ms	4,80 ms	4,28 ms
Pierwszy trening klasyfikatora	33,69 s	12,67 s	8,14 s

Generacja fałszywie pozytywnych okien	590,40 s	455,72 s	409,20 s
Drugi trening klasyfikatora	40,62 s	18,88 s	10,36 s
Analiza rozkładu błędu detekcji w zbiorze testowym	1938 s	1353 s	1228 s

Tablica 4.4: Czas wykonywania poszczególnych etapów badania metody HOG

4.1.7. Podsumowanie

Na podstawie uzyskanych wyników można stwierdzić, że najskuteczniejszą z przetestowanych wielkości komórki w deskrytorze HOG jest komórka 8x8 pikseli. Jako jedyna daje ona rozsądne wyniki przy obydwu walidacjach klasyfikatora. Również rozkład DET dla takiej wielkości komórki sugeruje wysoką skuteczność działania.

4.2. Macierz kowariancji cech

4.2.1. Algorytm obliczenia deskryptora

Algorytm obliczenia deskryptora cech, wykorzystującego macierz kowariancji ma następującą postać:

Dane: obraz $n \times m$ w przestrzeni barw RGB, k - rozmiar lokalnego bloku

Wynik: wektor deskryptora cech o długości $\lfloor \frac{m}{k} \rfloor \times \lfloor \frac{n}{k} \rfloor \times 6^2$

macierz próbek := [];

zamień przestrzeń barw RGB na skalę szarości;

podziel obraz na bloki $k \times k$;

foreach *blok* **do**

I_x := jednokrotny operator Sobela dla bloku w kierunku poziomym;

I_y := jednokrotny operator Sobela dla bloku w kierunku pionowym;

I_{xx} := dwukrotny operator Sobela dla bloku w kierunku poziomym;

I_{yy} := dwukrotny operator Sobela dla bloku w kierunku pionowym;

macierz cech := [];

foreach *piksel* \in *okno* **do**

wektor cech := $\left[|I_x| \quad |I_y| \quad |I_{xx}| \quad |I_{yy}| \quad \sqrt{I_x^2 + I_y^2} \quad \text{atan2}(I_y, I_x) \right]$;

dodaj wektor cech jako wiersz do macierzy cech;

end

oblicz macierz kowariancji macierzy cech;

połącz kolejne wiersze macierzy cech w jeden wektor;

dodaj wektor do macierzy próbek;

end

dokonaj normalizacji każdej próbki (w niniejszej pracy jest to norma L2);

połącz kolejne wiersze macierzy próbek w wektor;

return *wektor cech*

Algorytm 4: Algorytm obliczenia deskryptora cech, wykorzystującego macierz kowariancji

4.2.2. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji pierwszego klasyfikatora wytrenowanego próbkami, uzyskanymi na podstawie wyliczenia deskryptora cech, wykorzystującego macierz kowariancji, w zależności od rozmiaru bloku na obrazie, w którym liczono macierz kowariancji.

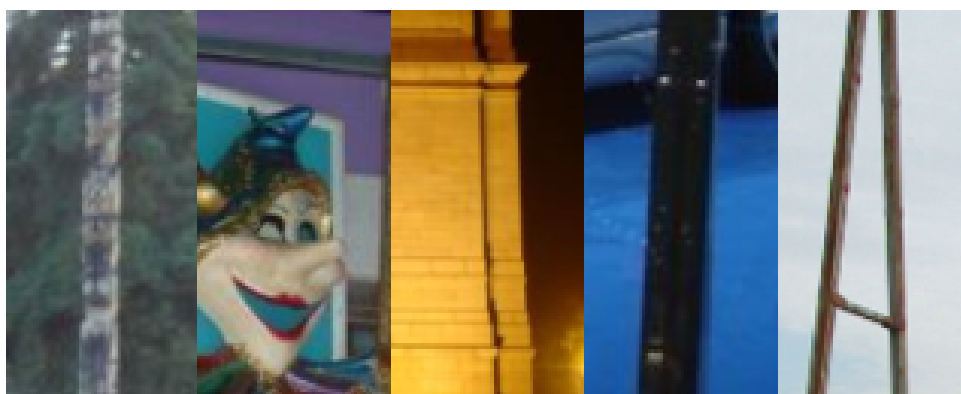
Rozmiar bloku	8x8	16x16	24x24
Funkcja jądra SVM	liniowa	liniowa	liniowa
Długość wektora cech	4608	832	360
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	599	593	584

Liczba negatywnych detekcji w pozytywnym zbiorze testowym	27	33	42
Celność metody w pozytywnym zbiorze testowym	95,69%	94,73%	93,29%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	21	30	29
Liczba negatywnych detekcji w negatywnym zbiorze testowym	885	876	877
Celność metody w negatywnym zbiorze testowym	97,68%	96,69%	96,80%

Tablica 4.5: Wyniki pierwszej walidacji krzyżowej przy użyciu deskryptora wykorzystującego macierz kowariancji cech

4.2.3. Przykłady fałszywych detekcji

Podobnie jak w przypadku metody HOG, zdecydowaną część zbioru fałszywie pozytywnych detekcji stanowiły okna, w których znajdowały się obiekty mające kształty przypominające ludzkie sylwetki, czyli np. obiekty zajmujące większość część okna, posiadające parę długich, równoległych krawędzi, jak słup czy drzewo. Ilości wygenerowanych fałszywie pozytywnych okien, w zależności od rozmiaru komórki, przedstawione zostały w tabeli 4.2.3. Przykładowe fałszywie pozytywne detekcje dla metody opierającej się na liczeniu macierzy kowariancji cech, zostały przedstawione na rysunku 4.3.



Rysunek 4.3: Przykłady fałszywie pozytywnych detekcji w metodzie korzystającej z macierzy kowariancji cech

Rozmiar komórki	8x8	16x16	24x24
Funkcja jądra SVM	liniowa	liniowa	liniowa

Ilość znalezionych pozytywnie fałszywych okien	4271	4141	7513
--	------	------	------

Tablica 4.6: Ilość znalezionych fałszywie pozytywnych okien przy użyciu deskryptora zbudowanego na podstawie macierzy kowariancji

4.2.4. Wyniki drugiej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji klasyfikatora po dotrenowaniu pierwszego klasyfikatora fałszywymi pozytywnymi detekcjami, wykrytymi w zbiorze testowym. W celu zminimalizowania efektu przetrenowania klasyfikatora wzięto pod uwagę 25% fałszywie pozytywnych detekcji wygenerowanych na zbiorze testowym.

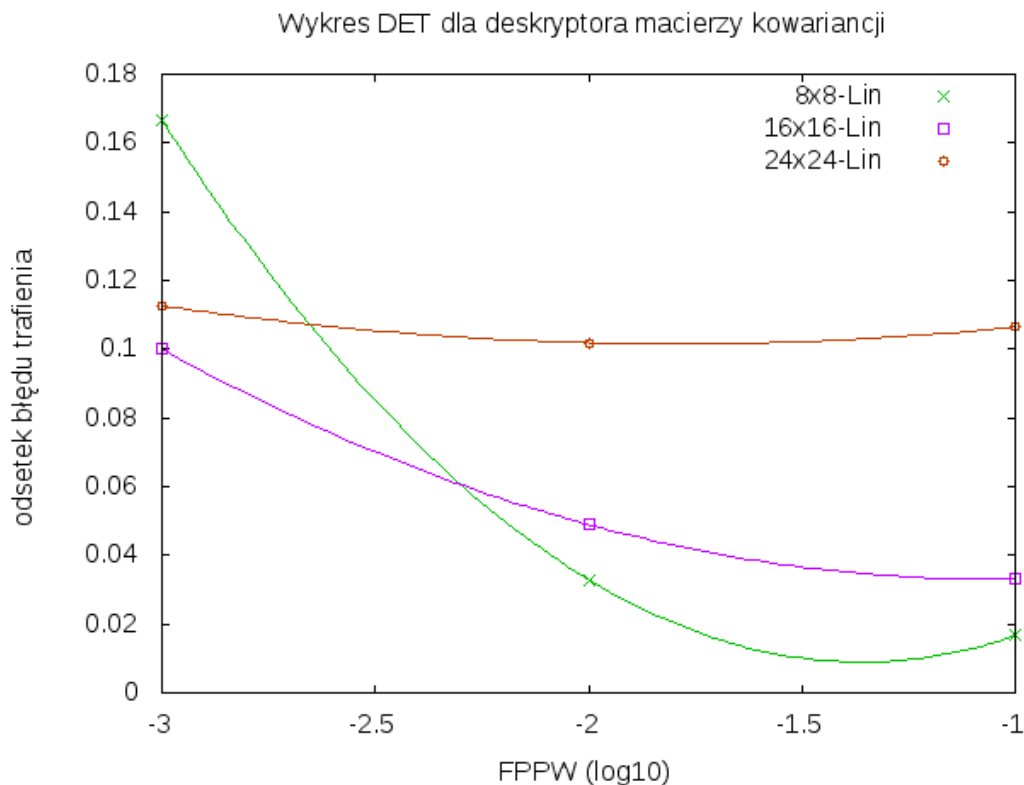
Rozmiar bloku	8x8	16x16	24x24
Funkcja jądra SVM	liniowa	liniowa	liniowa
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	571	502	289
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	55	124	337
Celność metody w pozytywnym zbiorze testowym	91,21 %	80,20 %	46,17 %
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	14	19	15
Liczba negatywnych detekcji w negatywnym zbiorze testowym	892	887	891
Celność metody w negatywnym zbiorze testowym	98,45 %	97,90 %	98,34 %

Tablica 4.7: Wyniki drugiej walidacji krzyżowej przy użyciu deskryptora wykorzystującego macierz kowariancji cech

4.2.5. Rozkłady błędów detekcji

Wykres rozkładu błędu detekcji zaprezentowany został na rysunku 4.4. Na podstawie wykresu można wyciągnąć wniosek, że w przypadku użycia liniowej funkcji jądra klasyfikatora SVM, najlepsze zachowanie prezentuje klasyfikator zbudowany przy użyciu macierzy kowariancji wyliczanej na blokach

16x16. Zachowuje on rozsądny wynik detekcji obiektu, gdy ma on do czynienia z obrazem, gdzie ilość FPPW jest rzędu 10^{-2} , a wraz z malejącą ilością fałszywych pozytywnych detekcji, odsetek błędu detekcji nie rośnie zbyt gwałtownie, jak np. w przypadku deskryptora zbudowanego o macierz kowariancji wyliczaną na blokach 8x8.



Rysunek 4.4: Rozkład błęd detekcji dla różnych przyjętych wielkości macierzy

4.2.6. Analiza złożoności poszczególnych kroków metody

W poniższej tabeli zestawiono czasy wykonywania poszczególnych kroków metodologii, w zależności od rozmiaru bloku, z którego wyliczana jest macierz kowariancji, na konfiguracji testowej.

Rozmiar bloku	8x8	16x16	24x24
Funkcja jądra SVM	liniowa	liniowa	liniowa
Ekstrakcja cech pozytywnych / próbka	11,12 ms	4,33 ms	3,51 ms
Ekstrakcja cech negatywnych / próbka	10,37 ms	3,54 ms	2,68 ms
Pierwszy trening klasyfikatora	50,81 s	10,75 s	5,99 s
Generacja fałszywie pozytywnych okien	909,12 s	296,06 s	231,88 s
Drugi trening klasyfikatora	68,49 s	14,41 s	8,79 s

Analiza rozkładu błędu detekcji w zbiorze testowym	3172 s	912 s	669 s
---	--------	-------	-------

Tablica 4.8: Czas wykonywania poszczególnych etapów badania metody opartej na macierzy kowariancji cech

4.2.7. Podsumowanie

Analizując powyższe wyniki można dojść do wniosku, że wielkościami bloków, dla których metoda daje rozsądne wyniki są bloki 8x8 pikseli i 16x16 pikseli. Użycie bloku 8x8 daje co prawda lepsze jakościowo wyniki w większości przypadków testowych, jednak jest to dużo bardziej kosztowne czasowo niż w przypadku bloku 16x16. Wynika to bezpośrednio z długości wektora cech, jaki powstaje w pierwszym przypadku. Deskryptor zbudowany na blokach o wielkości 24x24 pikseli jest złym deskryptorem i ma tendencję do szybkiego przetrenowania klasyfikatora.

4.3. LBP - Local Binary Patterns

4.3.1. Wprowadzenie

W pracy [MYL⁺08] zaprezentowano metodę budowy deskryptora cech opartego na oryginalnym algorytmie Local Binary Patterns, dającego jednak lepsze wyniki w problemach detekcji kształtów, niż oryginalna metoda, która pierwotnie opracowana została do detekcji tekstur. Wprawdzie pierwsze testy tej metody dawały obiecujące wyniki, jednak przy treningu klasyfikatora znacznie większymi zbiorami danych pokazały, że tak zbudowany deskryptor ma tendencję do przetrenowywania klasyfikatora. W związku z tym, w pracy zdecydowano się użyć tradycyjnej metody LBP.

4.3.2. Algorytm obliczenia deskryptora

Algorytm obliczenia deskryptora cech, wykorzystującego macierz kowariancji ma następującą postać:

Dane: obraz $n \times m$ w przestrzeni barw RGB, k - rozmiar lokalnego bloku, p - próg tworzenia wzorca

Wynik: wektor deskryptora cech o długości $\lfloor \frac{m}{k} \rfloor \times \lfloor \frac{n}{k} \rfloor \times 512$

macierz próbek := [];

zamień przestrzeń barw RGB na skalę szarości;

podziel obraz na bloki $k \times k$;

foreach *blok* **do**

w := zerowy wektor, 512-elementowy;

foreach *piksel* \in *blok* **do**

v := zerowy wektor, 8-elementowy;

foreach *sasiad* \in *wektor (po okręgu)* **do**

if $|sasiad - piksel| > p$ **then**

$v(sasiad) := 1$;

end

end

n := liczba odpowiadająca binarnej reprezentacji w ;

$w(n) := w(n) + 1$;

end

 dodaj wektor w do macierzy próbek;

end

dokonaj normalizacji każdej próbki (w niniejszej pracy jest to norma L2);

połącz kolejne wiersze macierzy próbek w wektor;

return *wektor cech*

Algorytm 5: Algorytm obliczenia deskryptora cech w metodzie LBP

4.3.3. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji pierwszego klasyfikatora wytrenowanego próbkami, uzyskanymi na podstawie wyliczenia deskryptora cech metodą LBP, w zależności od rozmiaru bloku na obrazie, w którym liczono histogram rozkładu wzorca.

Rozmiar bloku	32x32	64x64
Funkcja jądra SVM	liniowa	liniowa
Długość wektora cech	4096	1024
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	531	443
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	95	183
Celność metody w pozytywnym zbiorze testowym	84,82%	70,77%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	20	30
Liczba negatywnych detekcji w negatywnym zbiorze testowym	886	876
Celność metody w negatywnym zbiorze testowym	97,79%	96,69%

Tablica 4.9: Wyniki pierwszej walidacji krzyżowej przy użyciu deskryptora LBP

4.3.4. Przykłady fałszywych detekcji

Zbiór fałszywie pozytywnych detekcji uzyskanych w trakcie testowania metody LBP zawiera okna z różnymi skomplikowanymi kształtami, zawierającymi dużą ilość różnie zorientowanych krawędzi. Przeważnie obiekty te kontrastują dość wyraźnie z tłem. Podobieństwo wykrytych pozytywnie negatywnych okien nie jest tak oczywiste jak w przypadku poprzednich dwóch metod, ma to jednak podstawy w sposobie budowy deskryptora - opisuje on teksturę a nie kształt. Ilości wygenerowanych fałszywie pozytywnych okien, w zależności od rozmiaru bloku, przedstawione zostały w tabeli 4.3.4. Przykładowe fałszywie pozytywne detekcje dla metody LBP, zostały przedstawione na rysunku 4.5.

Rozmiar bloku	32x32	64x64
Funkcja jądra SVM	liniowa	liniowa
Ilość znalezionych pozytywnie fałszywych okien	4711	5292

Tablica 4.10: Ilość znalezionych fałszywie pozytywnych okien przy użyciu deskryptora zbudowanego na podstawie macierzy kowariancji

4.3.5. Wyniki drugiej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji klasyfikatora po dotrenowaniu pierwszego klasyfikatora fałszywymi pozytywnymi detekcjami, wykrytymi w zbiorze testowym. W celu zminimalizowania efektu przetrenowania klasyfikatora wzięto pod uwagę 25% fałszywie pozytywnych detekcji wygenerowanych na zbiorze testowym.

Rozmiar bloku	32x32	64x64
Funkcja jądra SVM	liniowa	liniowa
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	479	512
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	147	134
Celność metody w pozytywnym zbiorze testowym	76,52%	81,79%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	3	30
Liczba negatywnych detekcji w negatywnym zbiorze testowym	903	876
Celność metody w negatywnym zbiorze testowym	99,67%	96,69%

Tablica 4.11: Wyniki drugiej walidacji krzyżowej przy użyciu deskryptora LBP

4.3.6. Rozkłady błędów detekcji

Wykres rozkładu błędu detekcji zaprezentowany został na rysunku 4.6. Analizując wykres można dojść do wniosku, że oba przetestowane wielkości bloków, w których liczonych jest histogram metody LBP dają zbliżone do siebie rezultaty pod kątem celności detekcji.



Rysunek 4.5: Przykłady fałszywie pozytywnych detekcji w metodzie LBP

4.3.7. Analiza złożoności poszczególnych kroków metody

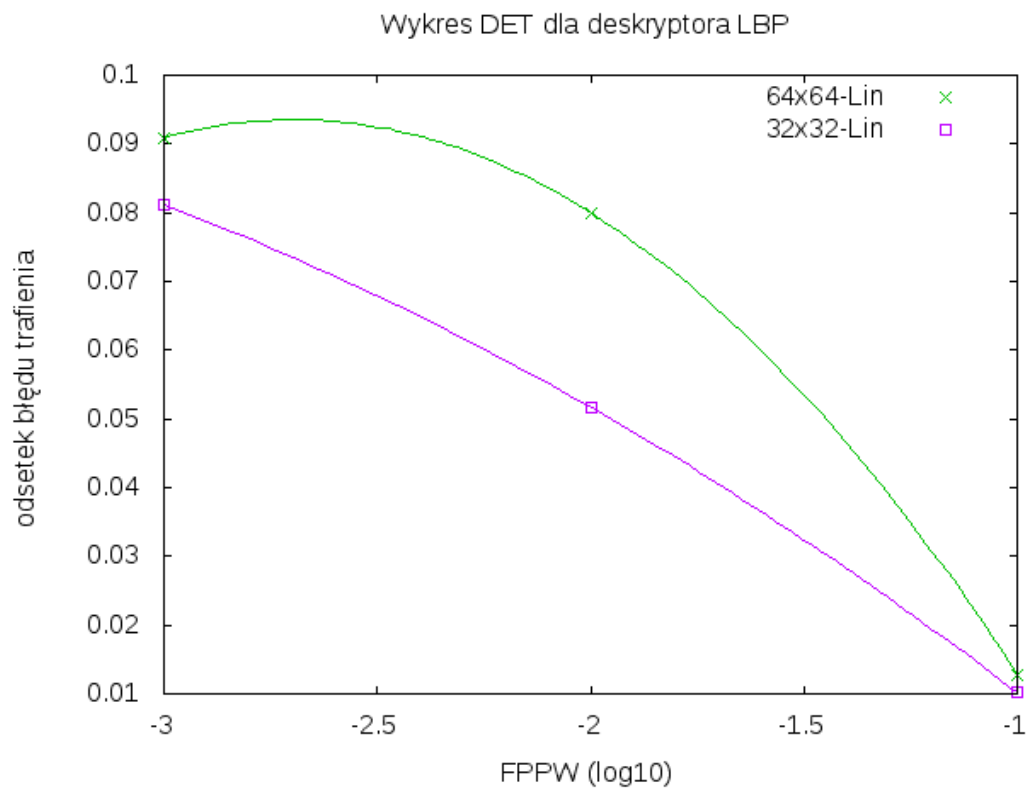
W poniższej tabeli zestawiono czasy wykonywania poszczególnych kroków metodologii, w zależności od rozmiaru bloku, w której wyliczany jest histogram metody LBP.

Rozmiar bloku	32x32	64x64
Funkcja jądra SVM	liniowa	liniowa
Ekstrakcja cech pozytywnych / próbka	14,10 ms	11,08 ms
Ekstrakcja cech negatywnych / próbka	12,42 ms	10,19 ms
Pierwszy trening klasyfikatora	79 s	27,67 s
Generacja fałszywie pozytywnych okien	917,78 s	657 s
Drugi trening klasyfikatora	89,83 s	31,62 s
Analiza rozkładu błędu detekcji w zbiorze testowym	4831 s	4914 s

Tabela 4.12: Czas wykonywania poszczególnych etapów badania metody LBP

4.3.8. Podsumowanie

Podsumowując przeprowadzone testy, pod względem jakościowym oba przetestowane rozmiary bloku, w którym na obrazie wyliczany jest histogram rozkładu wzorca, dają podobne rezultaty. W przypadku bloku 32x32 piksele jednak, poziom detekcji jest nieco wyższy w przypadku gdy rząd FPPW wynosi 10^{-2} . Biorąc jednak pod uwagę rozmiar wektora cech, jaki powstaje przy ich ekstrakcji, słuszniejszym wyborem wydaje się być blok o rozmiarze 64x64, gdzie długość wektora cech jest 4 mniejsza od bloku 32x32. W konsekwencji, trening SVM zajmuje krótszy czas.



Rysunek 4.6: Rozkład błędu detekcji dla różnych przyjętych wielkości bloków

4.4. Edgelets

4.4.1. Wprowadzenie

Algorytm zaproponowany przez autorów metody, B. Wu i R. Nevatię w pracy [WN05] nie jest typowym deskryptorem cech opisującym obraz w postaci wektora. Metoda zaproponowana przez autorów polega na budowie kaskadowego klasyfikatora, porównującego w każdym kroku podobieństwo fragmentu obrazu do każdego z wielu przygotowanych wcześniej wzorcowych kształtów. Ponieważ SVM, klasyfikator używany w tej pracy, jest mocnym klasyfikatorem, potrzebującym do treningu i klasyfikacji wektora zawierającego całość informacji, w niniejszej pracy dokonana została próba przystosowania metody do użycia właśnie z wykorzystaniem takiego rodzaju klasyfikatora.

Ze względu na bardzo wysoką złożoność metody, w przypadku porównywania z wieloma wzorcowymi kształtami, na podstawie zbioru treningowego, wygenerowano po 150 losowych masek dla fragmentów obrazu o rozmiarze 3x3, 4x4 i 5x5 i na ich podstawie dokonywano obliczenia deskryptora.

4.4.2. Algorytm obliczenia deskryptora

Proponowany algorytm obliczenia deskryptora cech, bazujący na oryginalnej metodzie *Edgelets* ma następującą postać:

Dane: obraz $n \times m$ w przestrzeni barw RGB, k - rozmiar maski, względem której dokonywane jest porównywanie, macierz wzorcowych masek ($l \times k^2$)

Wynik: wektor deskryptora cech o długości l

zamień przestrzeń barw RGB na skalę szarości;

oblicz gradient w obu kierunkach na obrazie np. używając maski $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$;

oblicz długość wektora gradientu w każdym pikselu;

dokonaj dyskretyzacji kąta gradientu w każdym pikselu: $\text{round}(\frac{\text{atan2}(I_y, I_x) * 6}{\pi})$;

$L := [1, 0.8, 0.5, 0, 0.5, 0.8]$;

foreach $\text{piksel} \in \text{obraz}$ **do**

foreach maska wzorcowa **do**

 ocena := 0;

foreach $\text{piksel} \in \text{maska}$ **do**

 ocena := ocena + $\sqrt{I_x^2 + I_y^2} * L[|\text{pikselobrazu} - \text{pikselmaski}| + 1]$;

end

 wektor(maska) := wektora(maska) + ocena;

end

end

return wektor cech

Algorytm 6: Proponowany algorytm deskryptora cech, wykorzystujący metodę *Edgelets*

4.4.3. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji klasyfikatora wytrenowanego próbkami, uzyskanymi na podstawie wyliczenia wyżej zaproponowanego deskryptora cech, w zależności od rozmiaru fragmentów obrazu, które były porównywane.

Rozmiar fragmentu	3x3	4x4	5x5
Funkcja jądra SVM	liniowa	liniowa	liniowa
Długość wektora cech	150	150	150
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	615	97	468
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	11	529	158
Celność metody w pozytywnym zbiorze testowym	98,24%	15,50%	74,76%
Liczba pozytywnych detekcji w negatywnym zbiorze testowym	650	362	476
Liczba negatywnych detekcji w negatywnym zbiorze testowym	256	544	430
Celność metody w negatywnym zbiorze testowym	28,26%	60,04%	47,46%

Tablica 4.13: Wyniki walidacji krzyżowej przy użyciu proponowanego deskryptora bazującego na metodzie *Edgelets*

4.4.4. Analiza złożoności poszczególnych kroków metody

Rozmiar fragmentu	3x3	4x4	5x5
Funkcja jądra SVM	liniowa	liniowa	liniowa
Ekstrakcja cech pozytywnych / próbka	118,31 ms	158,6 ms	114,40 ms
Ekstrakcja cech negatywnych / próbka	100,46 ms	161,40 ms	136,56 ms
Pierwszy trening klasyfikatora	0,67 s	1,29 s	0,98 s

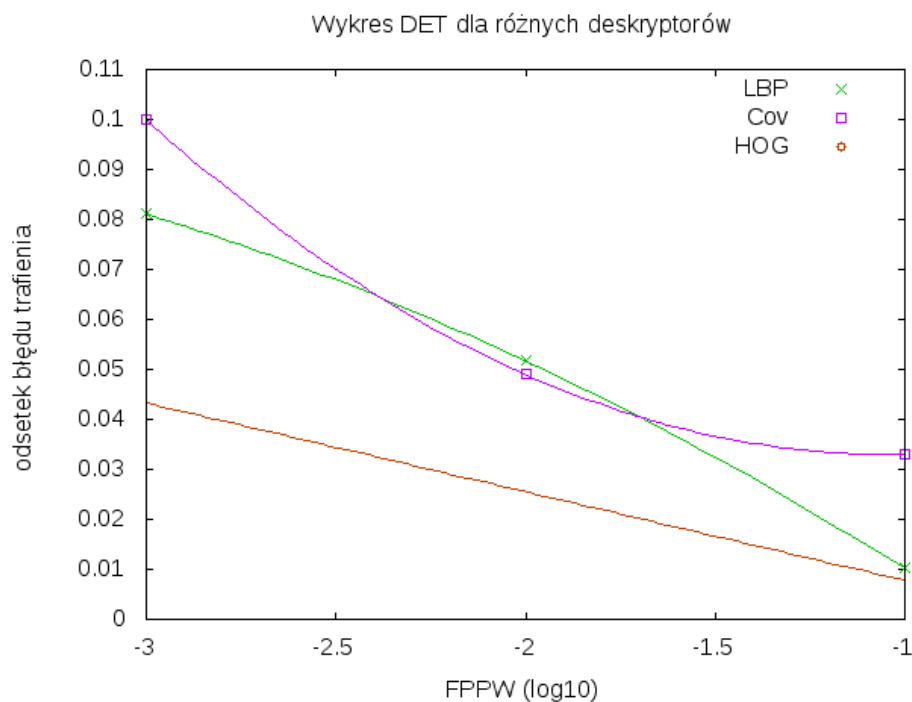
Tablica 4.14: Czas wykonywania poszczególnych etapów badania metody opartej na metodzie *Edgelets*

4.4.5. Podsumowanie

Uzyskanie nieobiecujących wyników w walidacji krzyżowej dla danej metody, a także jej bardzo duża złożoność poskutkowały zaniechaniem dalszych badań. Dodatkowo, informacje na podstawie jakich budowany jest wektor deskryptora (gradient obliczony na obrazie) brane są pod uwagę w innych, mniej złożonych i skuteczniejszych metodach takich jak HOG czy uzyskującej wektor cech z macierzy kowariancji. Takie podejście opisu obiektu zatem nie sprawdza się w przypadku, gdy do klasyfikacji używamy mocnych klasyfikatorów, takich jak Support Vector Machine.

4.5. Zestawienie przebadanych metod

Wspólny wykres, dla najlepszych wariantów każdej z metod przedstawiono na rysunku 4.7. Można z niego wywnioskować, że wszystkie z przebadanych metod, dla przyjętej w tej pracy metodologii, reprezentują podobny poziom detekcji dla FPPW tego samego rzędu. Najlepszą z przebadanych metod, w kontekście detekcji ludzi, bazując na wynikach rozkładu, jest metoda HOG. Należy jednak pamiętać, że metoda ta została zaprojektowana *stricte* dla detekcji sylwetek ludzkich na obrazie, podczas gdy dwie pozostałe są metodami bardziej uniwersalnymi.



Rysunek 4.7: Wspólny wykres DET dla przebadanych metod

5. Łączenie działania deskryptorów

5.1. Wprowadzenie

Ze względu na fakt, że każdy z elementów wektora traktowany jest przez klasyfikator jako zmienna niezależna od reszty, połączenie wynikowych wektorów dwóch lub więcej metod pozwoli na jeszcze dokładniejsze opisanie danego obiektu. Szczególną korzyść można odnieść w przypadku, gdy łączone ze sobą deskryptory dokonują ekstrakcji cech z oryginalnego obrazu stosując odmienne podejście, dzięki czemu w łączonych ze sobą wektorach nie ma powielonych tych samych informacji. Oczywiście kosztem potencjalnego poprawienia skuteczności metody, oprócz konieczności wyliczenia kilku wektorów cech, jest zwiększenie wymiarowości wynikowego wektora cech, co ciągnie za sobą dłuższy czas treningu klasyfikatora i klasyfikacji.

W przebadanych metodach metody HOG i macierzy kowariancji cech bazują na informacjach dotyczących krawędzi obrazu wejściowego. Odmienne podejście zastosowano w metodzie LBP, która dokonuje ekstrakcji informacji dotyczących tekstury na obrazie. Biorąc pod uwagę powyższe, przebadano zachowanie następujących połączeń deskryptorów: HOG i LBP oraz macierzy kowariancji cech i LBP.

5.2. Łączenie deskryptora kowariancji i LBP

5.2.1. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji pierwszego klasyfikatora wytrenowanego próbkami uzyskanymi w wyniku połączenia deskryptorów cech, wykorzystującego macierz kowariancji oraz LBP. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	Kowariancja	LBP	Połączenie
Długość wektora cech	832	4096	4928
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	593	531	601
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	33	95	25
Celność metody w pozytywnym zbiorze testowym	94,73%	84,82%	96,01%

Liczba pozytywnych detekcji w negatywnych zbiorze testowym	30	20	21
Liczba negatywnych detekcji w negatywnym zbiorze testowym	876	886	885
Celność metody w negatywnym zbiorze testowym	96,69%	97,79%	97,68%

Tablica 5.1: Wyniki pierwszej walidacji krzyżowej przy użyciu łączonego deskryptora wykorzystującego macierz kowariancji cech i LBP

5.2.2. Wyniki drugiej walidacji

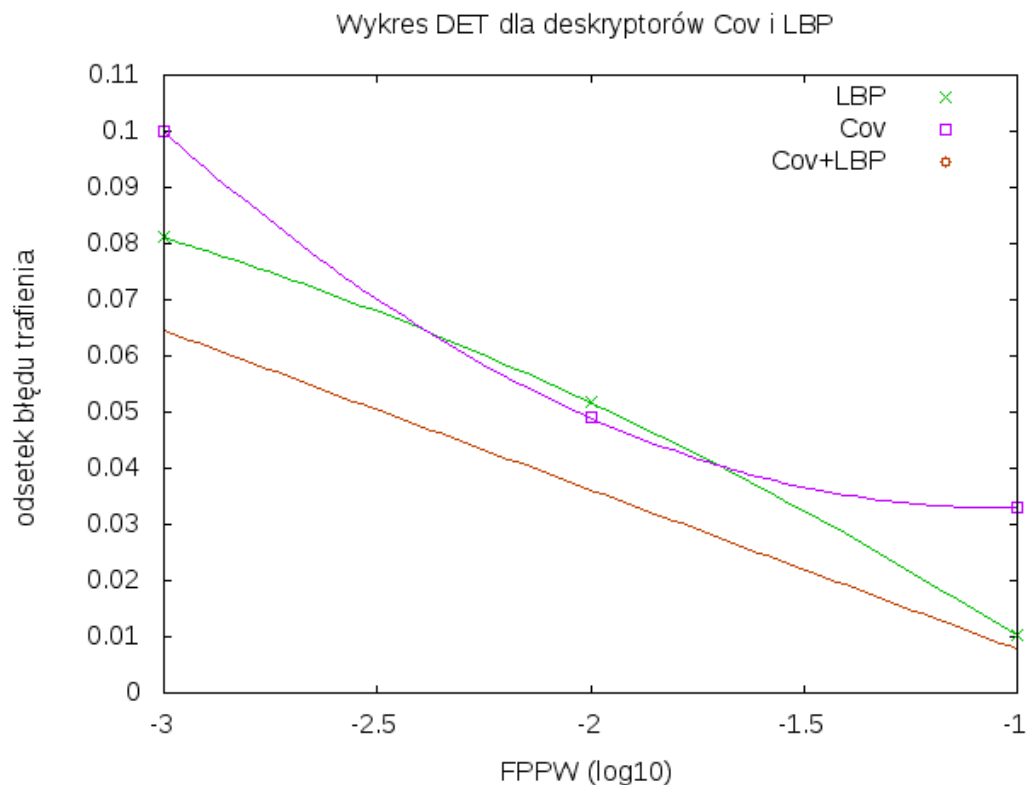
W poniższej tabeli zestawiono wyniki krzyżowej walidacji drugiego klasyfikatora po dotrenowaniu pierwszego klasyfikatora fałszywymi pozytywnymi detekcjami, wykrytymi w zbiorze testowym. W celu zminimalizowania efektu "przetrenowania" klasyfikatora wzięto pod uwagę 25% fałszywie pozytywnych detekcji wygenerowanych na zbiorze testowym. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	Kowariancja	LBP	Połączenie
Długość wektora cech	832	4096	4928
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	502	512	584
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	124	134	40
Celność metody w pozytywnym zbiorze testowym	80,20%	81,79%	93,29%
Liczba pozytywnych detekcji w negatywnych zbiorze testowym	19	30	17
Liczba negatywnych detekcji w negatywnym zbiorze testowym	887	876	889
Celność metody w negatywnym zbiorze testowym	97,90%	96,69%	98,12%

Tablica 5.2: Wyniki drugiej walidacji krzyżowej przy użyciu łączonego deskryptora wykorzystującego macierz kowariancji cech i LBP

5.2.3. Rozkłady błędów detekcji

Wykres rozkładu błędu detekcji przy połączeniu deskryptorów HOG i LBP zaprezentowany został na rysunku 5.1. Na podstawie wykresu można zauważyć, że połączenie tych metod nie poprawia działania klasyfikatora, a w kontekście zbioru testowego i metodologii, na którym został badany nawet nieco pogarsza.



Rysunek 5.1: Rozkład błędu detekcji dla połączenia metody LBP i kowariancji cech

5.2.4. Analiza złożoności poszczególnych kroków metody

W poniższej tabeli zestawiono czasy wykonywania poszczególnych kroków metodologii na konfiguracji testowej. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	Kowariancja	LBP	Połączenie
Ekstrakcja cech pozytywnych / próbka	4,33 ms	11,08 ms	14,72 ms
Ekstrakcja cech negatywnych / próbka	3,54 ms	10,19 ms	13,86 ms

Pierwszy trening klasyfikatora	10,75 s	27,67 s	57,67 s
Generacja fałszywie pozytywnych okien	296,06 s	657 s	884,73 s
Drugi trening klasyfikatora	14,41 s	31,62 s	68,62 s
Analiza rozkładu błędu detekcji w zbiorze testowym	3172 s	4914 s	5200 s

Tablica 5.3: Czas wykonywania poszczególnych etapów badania przy użyciu łączonego deskryptora wykorzystującego macierz kowariancji cech i LBP

5.2.5. Podsumowanie

Nowo utworzony deskryptor można, pod względem jakościowym, ocenić znacznie lepiej niż jego pojedyncze składowe, używane do detekcji osobno. Potwierdza to zarówno rozkład DET porównany z rozkładami dla składowych jak i wyniki krzyżowych walidacji, w szczególności po dotrenowaniu klasyfikatora. Odbywa się to jednak znacznie zwiększonych kosztem złożoności użycia nowej metody. Wynika to bezpośrednio z faktu, że klasyfikator operuje na znacznie większych wektorach cech, które powstają z połączenia dwóch.

5.3. Łączenie deskryptora HOG i LBP

5.3.1. Wyniki pierwszej walidacji

W poniższej tabeli zestawiono wyniki krzyżowej walidacji pierwszego klasyfikatora wytrenowanego próbkami uzyskanymi w wyniku połączenia deskryptorów cech, wykorzystującego deskryptory HOG oraz LBP. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	HOG	LBP	Połączenie
Długość wektora cech	7308	4096	11404
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	575	531	585
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	57	95	41
Celność metody w pozytywnym zbiorze testowym	91,85%	84,82%	93,45%

Liczba pozytywnych detekcji w negatywnych zbiorze testowym	31	20	21
Liczba negatywnych detekcji w negatywnym zbiorze testowym	875	886	885
Celność metody w negatywnym zbiorze testowym	96,58%	97,79%	97,68%

Tablica 5.4: Wyniki pierwszej walidacji krzyżowej przy użyciu łączonego deskryptora wykorzystującego deskryptory HOG i LBP

5.3.2. Wyniki drugiej walidacji

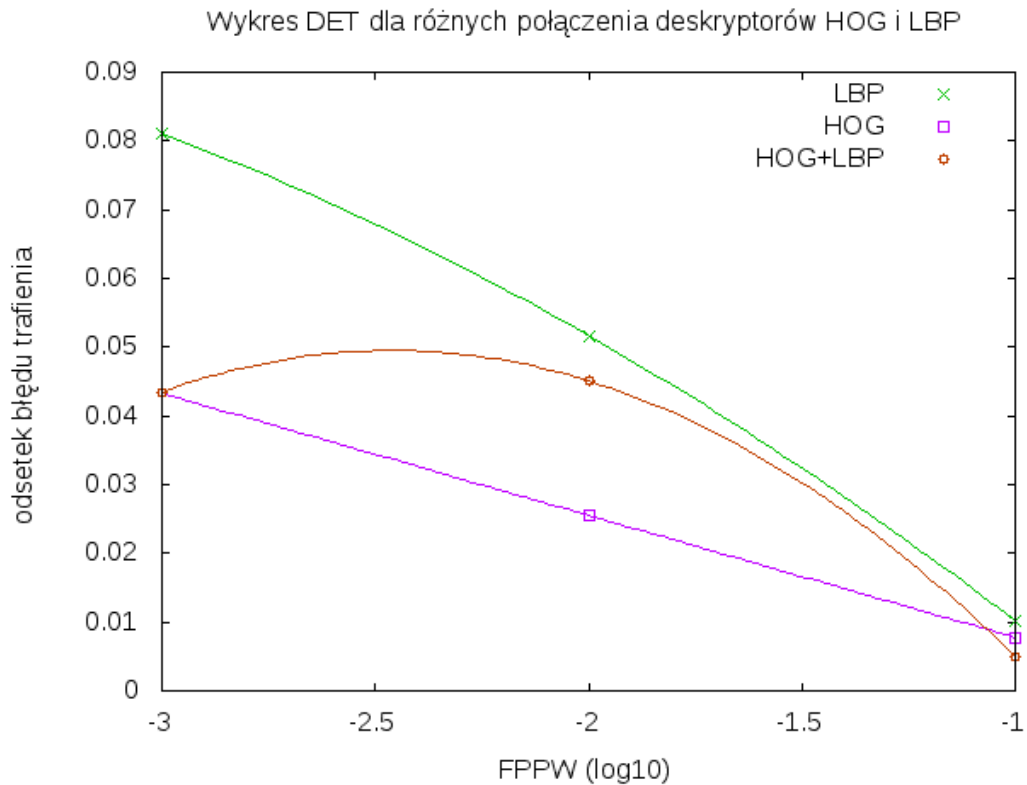
W poniższej tabeli zestawiono wyniki krzyżowej walidacji drugiego klasyfikatora po dotrenowaniu pierwszego klasyfikatora fałszywymi pozytywnymi detekcjami, wykrytymi w zbiorze testowym. W celu zminimalizowania efektu "przetrenowania" klasyfikatora wzięto pod uwagę 25% fałszywie pozytywnych detekcji wygenerowanych na zbiorze testowym. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	HOG	LBP	Połączenie
Długość wektora cech	7308	4096	11404
Liczba pozytywnych detekcji w pozytywnym zbiorze testowym	531	512	564
Liczba negatywnych detekcji w pozytywnym zbiorze testowym	95	134	60
Celność metody w pozytywnym zbiorze testowym	84,82%	81,79%	90,10%
Liczba pozytywnych detekcji w negatywnych zbiorze testowym	32	30	13
Liczba negatywnych detekcji w negatywnym zbiorze testowym	874	876	893
Celność metody w negatywnym zbiorze testowym	96,47%	96,69%	98,57%

Tablica 5.5: Wyniki drugiej walidacji krzyżowej przy użyciu łączonego deskryptora wykorzystującego deskryptory HOG i LBP

5.3.3. Rozkłady błędów detekcji

Wykres rozkładu błędu detekcji przy połączeniu deskryptorów HOG i LBP zaprezentowany został na rysunku 5.2. Na podstawie wykresu można zauważyć, że połączenie metod wyraźnie poprawia skuteczność działania metody dla takiego samego rzędu fałszywie pozytywnych detekcji na obrazie.



Rysunek 5.2: Rozkład błędu detekcji dla połączenia metody LBP i HOG

5.3.4. Analiza złożoności poszczególnych kroków metody

W poniższej tabeli zestawiono czasy wykonywania poszczególnych kroków metodologii na konfiguracji testowej. Dla porównania zestawiono te wyniki z wynikami uzyskanymi dla pojedynczych deskryptorów.

Metoda	HOG	LBP	Połączenie
Ekstrakcja cech pozytywnych / próbka	6,23 ms	11,08 ms	15,24 ms
Ekstrakcja cech negatywnych / próbka	5,27 ms	10,19 ms	14,05 ms
Pierwszy trening klasyfikatora	33,69 s	27,67 s	69,89 s
Generacja fałszywie pozytywnych okien	590,40 s	657 s	987,32 s

Drugi trening klasyfikatora	40,62 s	31,62	82,39 s
Analiza rozkładu błędu detekcji w zbiorze testowym	1938 s	4914 s	5730 s

Tablica 5.6: Czas wykonywania poszczególnych etapów badania przy użyciu łączonego deskryptora wykorzystującego macierz kowariancji cech i LBP

5.3.5. Podsumowanie

Biorąc pod uwagę wyniki walidacji krzyżowej, z której można wywnioskować nieznaczną poprawę działania deskryptora HOG, a także wykres rozkładu błędu detekcji, który sugeruje negatywny wpływ połączenia, można dojść do wniosku, że w kontekście detekcji sylwetek ludzkich deskryptor LBP w słabym stopniu uzupełnia metodę HOG.

6. Podsumowanie

W niniejszej pracy przeprowadzono badania literatury naukowej pod kątem wybranych metod tworzenia deskryptorów cech używanych w metodach detekcji obiektów. Ponadto, zaprezentowano kompletną metodologię służącą do przebadania, zarówno pod kątem jakościowym jak i złożoności, danej metody - począwszy od generacji negatywnego zbioru testowego, przez ekstrakcję cech, budowę pierwszego i dotrenowanego klasyfikatora, walidację klasyfikatora, aż po budowę rozkładu błędu detekcji, który jest pomocnym narzędziem przy wyborze metody detekcji do konkretnego zastosowania.

Korzystając z powyższej metodologii przebadano trzy z czterech przeanalizowanych metod i porównano je. Dokonano także próby przystosowania ostatniej z nich, metody *edgelets* do metodologii, w której do klasyfikacji używany jest mocny klasyfikator, jak SVM.

Oprócz tego przebadano, korzystając z tej samej metodologii, efekt połączenia ze sobą dwóch metod. Jak wynika z przeprowadzonych badań, łączenie działania deskryptorów pozwala na polepszenie skuteczności detekcji, gdy połączone deskryptory zawierają niepokrywające się ze sobą informacje oraz metody te nie są dedykowane konkretnemu zastosowaniu. Takie ulepszenie odbywa się jednak kosztem złożoności czasowej.

W dalszej pracy warto dokonać próby budowy metody, podobnej do metody *edgelets*, jednak potrafiącej dobrze opisywać obraz wejściowy przy pomocy niezbyt długiego wektora cech. Można również zastanowić się nad znalezieniem deskryptora bazującego na metodzie LBP, jednak posiadającego lepsze właściwości opisu orientacji i długości krawędzi na obrazie. Próby takiej dokonali autorzy pracy [ZP07], jednak w bardziej wymagających testach metoda ta okazuje się zawodna, prawdopodobnie poprzez zbyt ogólny deskryptor. W dalszej pracy warto również przeprowadzić badania prowadzące do możliwości optymalizacji zaprezentowanych metod i algorytmów pod kątem zrównoleglenia wykonywania poszczególnych operacji prowadzących do obliczenia wartościżądanego deskryptora cech. Zbadanie możliwości wykonywania części lub całości poszczególnych algorytmów na wielu rdzeniach jednocześnie pozwoli na zastanowienie się nad możliwością ich przyspieszenia w rzeczywistych zastosowaniach.

Bibliografia

- [AMH⁺06] Timo Ahonen, Student Member, Abdenour Hadid, Matti Pietikäinen, and Senior Member. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:2037–2041, 2006.
- [Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.
- [DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.
- [FR94] William T. Freeman and Michal Roth. Orientation histograms for hand gesture recognition. Technical report, Mitsubishi Electric Research Labs., 201, 1994.
- [GC11] Guolong Gan and Jian Cheng. Pedestrian detection based on hog-lbp feature. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pages 1184 –1187, dec. 2011.
- [Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [MDK⁺97] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. pages 1895–1898, 1997.
- [MYL⁺08] Yadong Mu, Shuicheng Yan, Yi Liu, T. Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.
- [OPH96] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

- [TPM06] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *In Proc. 9th European Conf. on Computer Vision*, pages 589–600, 2006.
- [WN05] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, pages 90–97, Washington, DC, USA, 2005. IEEE Computer Society.
- [ZP07] Guoying Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915 –928, june 2007.