

POLITECHNIKA WROCLAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka

SPECJALNOŚĆ: Technologie informacyjne w systemach automatyki

PRACA DYPLOMOWA MAGISTERSKA

Badanie metod rozpoznawania znaków
drogowych z wykorzystaniem wielu źródeł
obrazu

Research on traffic sign recognition methods
using multiple image source

autor : BARTOSZ LENARTOWICZ

Opiekun pracy:
Dr inż. Bartosz Jabłoński W04/K8

OCENA PRACY:

WROCLAW 2019

Streszczenie

[here] (streszczenie na sam koniec)

Spis treści

Streszczenie	2
Analiza problemu, cel i zakres pracy	2
Publiczne bazy danych znaków	4
Implementacja	4
Metody detekcji obiektów na obrazach panoramicznych.....	4
Algorytmy łączenia obrazów w panoramę	4
Metody klasyczne	5
Metody bazujące na punktach charakterystycznych (kluczowych)	6
Łączenie obrazów	8
Segmentacja obrazów	8
Wykrywanie na podstawie koloru	9
Wykrywanie na podstawie kształtu	10
Wykrywanie na podstawie tekstury	12
Podejście hybrydowe	13
Inne podejścia	14
Wnioski	15
Klasyfikacja znaków drogowych	15
Sieci neuronowe	16
SVM.....	18
GA	18
K-d Tree.....	19
Podsumowanie	19
Program	19
Propozycja rozwinięcia/konstrukcji metody	19
Przygotowanie środowiska badawczego i plan badań	19
Podsumowanie	19
Bibliografia.....	19

Analiza problemu, cel i zakres pracy

Rozpoznanie znaków drogowych przez algorytmy przetwarzania obrazów jest aspektem niosącym ze sobą wiele korzyści, szczególnie jeżeli analiza następuje w czasie rzeczywistym.

Pierwszą korzyścią jaka się nasuwa jest wspomaganie kierowców podczas jazdy. Po nieco dłuższym zastanowieniu można wywnioskować, że rozpoznawanie znaków jest również istotną kwestią dla pojazdów. Dodatkowo rozpoznanie znaków drogowych może realnie przyczynić się do polepszania bezpieczeństwa na drodze. Przykładowo po rozpoznaniu znaku drogowego można zbadać w jakim stopniu jest on czytelny przez porównanie z wzorcem (np. czy nie został zamazany, zakrzywiony) i jeżeli tak się stało poinformować o tym służbę drogową. Istnieje szereg dziedzin gdzie analiza znaków drogowych może wprowadzić wiele innowacji. Prawdopodobnie dlatego inżynierowie od ponad dwóch dekad starają się sprostać temu zadaniu. [kiedy pierwsza próba]

Starając się wyjaśnić zakres pracy magisterskiej należy również rozszerzyć kilka terminów, które zostały w niej poruszone. W literaturze problem rozpoznawania znaków drogowych przedstawia się jako problem TSR (ang. Traffic Sign Recognition). Jest on składową większego zagadnienia jakim jest analiza sytuacji w oku samochodu podczas jazdy, a czasem nawet wewnątrz niego. W takim przypadku obszar wokół samochodu traktowany jest jako scena w, której należy rozpoznawać i analizować obiekty różnego typu. Zostaje to wspomniane, ponieważ niektóre systemy prócz rozpoznania znaków drogowych są w stanie rozpoznawać obiekty takie jak samochody, rowerzystów i pieszych. Taki system został przedstawiony w pracy [1]. Dodatkową zaletą pewnych systemów jest możliwość obiektów np. w celu poinformowania kierowców o kursie kolizyjnym oraz zapobiegnięcie wykrycia kilkakrotnie tego samego obiektu. Według autorów pracy [2] ostatni aspekt jest szczególnie ważny dla systemów w których pojazd jest kierowany przez kierowcę (a nie przez algorytm). Wielokrotne informowanie kierowcy o znaku może rozproszyć jego uwagę powodując tym samym zagrożenie. Autor w swojej pracy wspomina, że traktowanie kierowcy jako integralną część środowiska może polepszyć parametry systemu.

Chcąc zrealizować w skończonym czasie temat pracy należało wprowadzić pewne ograniczenia. Pierwszym z nich jest problem wykrywania obiektów zawężony do problemu TSR. Dodatkowo w pracy główny nacisk został położony na wykrywanie znaków pionowych. Obrazy zostały zarejestrowane przez kamery cyfrowe obejmujące obszar przed samochodem [oraz części prawego pobocza?]. Klatki utrwalone w tym samym czasie zostały połączone w obraz panoramiczny, na którym wystąpiła detekcja obiektów.

Skupiając się na problemie badawczym czyli wpływie zastosowania kilku kamer, zamiast kamery pojedynczej, należało przyjąć kryteria oceny. Problem oceny został ujęty jako zagadnienie statystyczne. Hipotezą zerową było stwierdzenie, że w przeszukiwanym obszarze nie ma znaków drogowych. Z pozoru nielogiczne założenie wprowadziło kilka udogodnień w rozumowaniu. Błędami pierwszego rodzaju w takim przypadku było wykrycie znaków drogowych gdzie w rzeczywistości ich nie ma. Błędem drugiego rodzaju było ominięcie znaku drogowego na obszarze gdzie w rzeczywistości on występuje. W pracy została przebadana skuteczność rozpoznawania znaków po wprowadzeniu dodatkowych kamer, dlatego mniejsze znaczenie przypisano czasom trwania poszczególnych algorytmów.

Warto zwrócić też uwagę na fakt, że system, który jest w stanie wykryć kilka różnych typów znaków lecz jest obciążony dużym błędem wykrywalności, w niektórych zastosowaniach może być uważany za lepszy niż system, który może wykryć tylko jeden określony typ znaku, ale robi to bardzo dobrze. Oczywiście możliwa jest również sytuacja odwrotna.

Publiczne bazy danych znaków

Ważnym aspektem do testowania algorytmów jest ujednolicona baza danych, ponieważ dopiero na jednolitej bazie danych można określić skuteczność algorytmu. Istnieje kilka publicznie dostępnych zestawów danych znaków drogowych:

1. Niemiecki test TSR (GTSRB),
2. KUL Belgium Zestaw danych znaków drogowych (zestaw danych KUL),
3. Szwedzki zestaw danych znaków drogowych (zestaw danych STS),
4. Baza danych obrazów RUG Traffic Sign (zestaw danych RUG),
5. Baza danych Stereopolis

Najważniejszą bazą danych z pozycji tej pracy jest baza danych GTSRB, ponieważ większość algorytmów była właśnie testowana na tej bazie danych.

Implementacja

Hardwareowa implementacja nie jest przedmiotem tej pracy natomiast jest niezbędna do jej realizacji przy próbie wdrożenia systemu. Autorzy artykułu [3] proponują swoje rozwiązanie implementacyjne [here](opisać dalej)

Metody detekcji obiektów na obrazach panoramicznych

Problem identyfikacji obiektów na obrazach panoramicznych pociąga za sobą szereg zagadnień do zrealizowania. Pierwszym z nich jest stworzenie obrazu panoramicznego z kilku ujęć otoczenia. Po stworzeniu panoramy należy rozpoznać co znajduje się na obrazie. Zagadnienie rozpoznania obrazu w większości przypadków realizowane jest w trzech krokach. Pierwszym krokiem jest przetwarzanie wstępne tj. nisko-poziomowe. Podczas takiego przetwarzania następuje np. akwizycja obrazu i w ogólności przetwarzanie-nisko-poziomowe ma na celu poprawę jakości obrazu poprzez eliminację zakłóceń, poprawę kontrastu, filtrację itp. Kolejny krok nazwany przetwarzaniem średniego poziomu i dotyczy segmentacji obrazu. Podczas segmentacji wydzielone zostają cechy obrazu. Przykładowo znajduje się punkty kluczowe lub obszary, w których mogą wystąpić obiekty. Po segmentacji następuje próba rozpoznania i klasyfikacji obiektu. Bardziej zaawansowane metody pracujące na filmach posiadają również etap śledzenia wykrytych obiektów. Ponieważ każdy etapów przetwarzania obrazów można traktować jako oddzielny problem, poniżej w osobnych podrozdziałach zostały przedstawione metody rozwiązania każdego zagadnienia.

Algorytmy łączenia obrazów w panoramę

Istnieje szereg algorytmów starających się sprostać problemowi dopasowania kilku obrazów do siebie. Wiele algorytmów jest odporne na duże rotacje, translacje lub różną skalę obrazów wejściowych. Istotną cechą poszukiwanego algorytmu powinna być odporność na różnicę w jasności w obrazach wejściowych oraz szumy spowodowane drganiami i rotacją oraz zabrudzeniami typu: błoto, kurz, krople wody.

W wielu pracach łączenie obrazów w panoramę zostaje podzielone na kilka etapów. Pierwszym z nich jest faza dopasowania obrazów do siebie. W tej fazie poprzez ekstrakcje cech szczególnych klatki wyłania się punkty kluczowe, niezmiennicze służące do dopasowania obrazów. Warto zaznaczyć, że takie punkty mogą posłużyć w kolejnych etapach do

identyfikacji obiektów. Następnie znajduje się najlepsze dopasowanie obrazów do siebie. Po dopasowaniu obrazów następuje fuzja właściwa powodująca scalenie obrazu w panoramę.

Poniżej zostały omówione różne algorytmy tworzenia panoramy. Wpierw zostały zaprezentowane metody klasyczne, dziś raczej już o znaczeniu historycznym. Następnie zaprezentowano kilka metod bazujących na punktach kluczowych, charakterystycznych. Są to metody najpopularniejsze mające największe znaczenie w rzeczywistości.

Metody klasyczne

Jak zostało wspomniane badanie algorytmów dopasowania obrazów należy zacząć od metod klasycznych. Metody te dopasowują obrazy do siebie poprzez wyznaczenie maksimum miary dopasowania. Dopasowanie wykonuje się głównie metodami optymalizacji funkcji. Metody nie zostały omówione szczegółowo ponieważ nie są używane w pracy. Do metod klasycznych optymalizacji funkcji można zaliczyć metodę Powella z użyciem metody Brenta [4]. Innym podejściem wyznaczenia zbieżności funkcji jest metoda Gaussa-Newtona, Levenberga-Marquardta [5].

Template matching

Klasyczną metodą dopasowania dwóch klatek do siebie jest metoda template matching. Metoda pozwana na znajdowanie na obrazie jego wycinka (wzorca). Taki algorytm jako obraz główny przyjmuje jeden z obrazów, a następnie jako wzorzec przyjmuje się wycinek z obrazu drugiego. Dla poprawienia działania przyjmuje się kilka wycinków obrazów. Ponieważ została wyznaczona lokalizacja wzorca na obrazie głównym oraz znane jest położenie wyciętego elementu na drugim obrazie można statystycznie dopasować obrazy. Najpopularniejszą używaną miarą dopasowania jest suma kwadratów błędów. Metoda template matching została przedstawiona np. w pracy [6]. Metoda jednak posiada dużą złożoność obliczeniową oraz jest słabo odporna na rotację i szumy. Do metod klasycznych można zaliczyć również metody bazujące na informacji wzajemnej.

Szybki algorytm dopasowania obrazów dla potrzeb fuzji w czasie rzeczywistym

W pracy [7] opisano „Szybki algorytm dopasowania obrazów dla potrzeb fuzji w czasie rzeczywistym”. Algorytm jest szybszy od metod bazujących na optymalizacji funkcji oraz template matching. Został zaproponowany do łączenia klatek filmowych tego samego obszaru nagrywanych różnymi kamerami (autorzy testowali algorytm na dwóch kamerach – TV i IR), jednak może być stosowany do tworzenia panoramy. Algorytm został przedstawiony następująco. Początkowa faza polega na przeskalowaniu obrazów do tych samych rozmiarów. Oba obrazy zostają poddane działaniu filtru krawędziowego. Następnie pionowe wycinki jednego z obrazów stara się dopasować do drugiego. Najlepsze dopasowanie, identycznie jak w metodach klasycznych, wyznacza się za pomocą sumy najmniejszych kwadratów lub sumy modułów różnic. Następnie wybiera się za pomocą metody statycznej tj. średnia, mediana lub dominanta najlepsze dopasowanie ogólne i zostaje wyznaczona wynikowa pozioma translacja. Dominanta daje najlepsze wyniki dopasowania. Obrazy zostają przesunięte o wyznaczoną wartość i proceder powtarza się w pionie. Algorytm jest słabo odporny na rotację lecz jego największą zaletą jest szybkość.

Transformata Fouriera

Metody opierające się na przekształceniu Fouriera działają w dziedzinie częstotliwości i są znacznie szybsze od metod klasycznych. Podstawową metodą wykorzystującą przekształcenie Fouriera jest metoda Fouriera-Mellina. Wykorzystuje ona twierdzenie Fouriera o przesunięciu

i pozwala metodą korelacji fazowej wyznaczyć przesunięcie dwóch obrazów względem siebie. Metoda ma liniową złożoność obliczeniową i dobrze radzi sobie z rotacją obrazów.

Warta uwagi jest również metoda [8]. Polega podzieleniu dwóch obrazów na niewielkie obszary i dopasowaniu obrazów do siebie za pomocą szybkich, mało dokładnych metod z wykorzystaniem miary statystycznej korelacji. Następnie używając metod bardziej złożonych obliczeniowo ale dokładniejszych łączy się segmenty w całość i znajduje najlepsze globalne dopasowanie. Autorzy algorytmu w opisie przedstawiają, że algorytm wykorzystuje metodę quasi-Newtona, z piramidą obszarów. Takie łączenie obrazów pozwala na dużą eliminację błędów oraz jest bardzo szybkie.

Metody bazujące na punktach charakterystycznych (kluczowych)

Metody łączenia obrazów, które posiadają większą odporność na zakłócenia bazują na ekstrakcji cech. Cechy takie rozumiane są jako specyficzne konfiguracje pikseli układające się w struktury. Przykładami takich konfiguracji mogą być: linie, zakończenia linii, krawędzie lub kąty. Wykrywanie struktur zazwyczaj następuje za pomocą filtracji obrazu. Fragmenty obrazu, które nie zmieniają się podczas przekształceń obrazu nazywane są punktami kluczowymi. Ważnym elementem punktów kluczowych jest ich niezmiennosc względem siebie. W ogólności można stwierdzić, że punkty charakterystyczne są skalo-niezmiennicze. Oznacza to, że po wykryciu kilku punktów kluczowych/charakterystycznych na dwóch obrazach, można takie obrazy do siebie dopasować lub śledzić jakiś obiekt.

Detektor Harrisa

Jako punkty charakterystyczne można wykorzystać narożniki wykryte na obrazach. Detektor Harrisa, który jest ulepszoną wersją detektora Moraveca z powodzeniem wykrywa narożniki. Koncepcja detektora Moraveca polega na przeszukiwaniu obrazu z wykorzystaniem okna przeszukiwania. Gdy podczas przeszukiwania w jednym kierunku zostanie zauważona duża zmiana w jasności pikseli obszar zostaje zakwalifikowany jako krawędź. Jeżeli zmiana zostanie również zauważona w kierunku prostopadłym to obszar kwalifikuje się jako narożnik. Zmiany intensywności sprawdza się co 45° co jest istotną wadą. W detektorze Moraveca wadami są również zaszumiona odpowiedź z uwagi na binarną funkcję okna oraz minimum jako kryterium.

Chcąc poprawić algorytm Chris Harris w 1988 roku zaproponował ulepszoną wersję algorytmu z powodzeniem stosowaną do dnia dzisiejszego. W detektorze Harrisa wykorzystywana jest macierz autokorelacji w postaci:

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}} & \left(\frac{\partial I}{\partial x}\right)^2 \left(\frac{\partial I}{\partial y}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}} \\ \left(\frac{\partial I}{\partial x}\right)^2 \left(\frac{\partial I}{\partial y}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}} & \left(\frac{\partial I}{\partial y}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}} \end{bmatrix}$$

I jest funkcją intensywności, a σ jest odchyleniem standardowym funkcji Gaussa. Jeżeli różnica między wyznacznikiem macierzy M , a kwadratem jej śladu przemnożonym przez stałą k ($P(x, y) = \det(M) - k(\text{trace}(M))^2$) będzie mniejsza od zera $P(x, y) < 0$ to punkt $P(x, y)$ jest uważany za krawędź. Jeżeli zaś $P(x, y) > 0$ to punkt $P(x, y)$ uważany jest za narożnik. Jeżeli punkt jest bliski zera $P(x, y) \sim 0$ to uważa się, że obszar nie posiada znaczących zmian. Dodatkowo dla większej dokładności algorytmu wykrywania narożników wprowadza się próg $P(x, y) > T$. Zaleca się również wybrania punktów z lokalnym maksimum. Detektor Harrisa

wykrywa narożniki z wiele większą dokładnością niż detektor Moraveca oraz sprawdza narożniki pod każdym kątem. Algorytm może z powodzeniem zostać wykorzystywany do tworzenia panoram.

GFTT

Algorytm GFTT (ang. - Good Features To Track) jest kolejnym ulepszeniem jakie wprowadzili Jianbo Shi oraz Carlo Tomasi w 1994 roku dla detektora Harrisa. Na potrzeby algorytmu wymaga się by obraz wejściowy był czarno biały. Zmianie w odniesieniu do oryginalnego algorytmu polegała na zastosowaniu innego wzoru wyliczającego funkcje oceniającą. Zamiast jak w oryginale $P(x, y) = \det(M) - k(\text{trace}(M))^2$ w przypadku GFTT funkcja ta ma postać

$$P(x, y) = \min\left(\left(\frac{\partial I}{\partial x}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}}, \left(\frac{\partial I}{\partial y}\right)^2 e^{\frac{-(x^2+y^2)}{2\sigma^2}}\right)$$

Dzięki swojej modyfikacji algorytm znajduje najbardziej użyteczne wierzchołki obrazu. Każdy wykryty wierzchołek sprawdzana się czy przekroczył pewien próg zmiany jasności i jeżeli tego nie zrobił zostaje automatycznie odrzucony. Dodatkowo określa się minimalny dystans pomiędzy wierzchołkami. Kryteria pozwalają wyłonić najlepiej procentujące wierzchołki, czyli interesujące punkty kluczowe.

FAST

Detektor FAST (ang. Features from Accelerated Segment Test) w swoim kryterium porównuje jasność piksela z jasnością pikseli oddalonych o ustalony promień. W przypadku gdy jasność większości pikseli będzie się różnić od piksela centralnego o określoną wartość to punkt kwalifikowany jest jako narożnik. Przykładowo dla promienia równego 3 piksele jeżeli 12 z 16 pikseli będą jaśniejsze niż piksel centralny pomniejszony o pewną wartość to algorytm wykrył narożnik. Na potrzeby algorytmu promień powinien być dostatecznie mały (od 2 do 6 pikseli) oraz wartość progu kwalifikacji i ilości pikseli odpowiednio przeskalowana do promienia. Algorytm FAST osiąga przeciętne wyniki w wykrywaniu punktów kluczowych. Jego zaletą jest szybkość.

SIFT

Algorytmem który wykorzystuje punkty charakterystyczne na którego należało by zwrócić szczególną uwagę jest zaproponowany przez David Lowe algorytm SIFT (ang. Scale Invariant Feature Transform) [9]. Jest to jeden z bardziej popularnych algorytmów dlatego jego działanie zostało przybliżone w tym akapicie. Jak większość opisywanych algorytmów tak i ten prócz tworzenia panoramy z powodzeniem można zastosować do rozpoznawania obiektów. Algorytm SIFT realizowany jest w czterech krokach. Kroki zostały wpięrw przedstawione, a następnie szczegółowo opisane. Pierwszy krok nazwany „scale space extrema detection” wykrywa punkty ekstremalne na dwóch obrazach. W kroku drugim „accurate keypoint location” następuje dokładna lokalizacja punktów charakterystycznych. Następnie przypisuje się orientację wykrytym punktom w przestrzeni. Krok nazwany jest „keypoint orientation assignment”. Czwarty etap zawiera tworzenie deskryptorów dla punktów charakterystycznych. Szczegółowe omawianie algorytmu należy zacząć od procesu skalowania obrazu wejściowego do różnych wielkości. Dla każdego obrazu w skali zostaje użyty filtr Laplace’a, który pozwala uzyskać kontury obiektów D_n , obliczany za pomocą odjęcia dwóch obrazów rozmytych filtrem Gaussa z różnymi parametrem σ . Zazwyczaj większe rozmycie obrazu tworzy się

poprzez zwiększenie potęgi do której zostanie podniesiona stała k . Dla ułatwienia zrozumienia poniżej opisano wzór

$$D_n = \frac{1}{2\pi(\sigma k^{n+1})^2} e^{\frac{-(x^2+y^2)}{2k^2\sigma^2}} \cdot P(x, y) - \frac{1}{2\pi(\sigma k^n)^2} e^{\frac{-(x^2+y^2)}{2k^2\sigma^2}} \cdot P(x, y)$$

Zazwyczaj w za parametr odchylenia standardowego przyjmuje się $\sigma = 1,6$. Stała $k = \sqrt{2}$ podnoszona do kolejnych potęg. Następnie przez binaryzację wykrywa się lokalne maksima i minima. Taka operacja powoduje wykrycie ogromnej ilości punktów ekstremalnych dlatego należy zastosować dwa kryterium, które odfiltrują najlepsze możliwe punkty charakterystyczne. Pierwsze kryterium odrzuca płytkie minima bądź niewielkie maksima za pomocą rozwinięcia funkcji w szereg Taylora. Dzięki temu odrzucone punkty, które powstały na obszarze np. nieba. Drugie kryterium sprawdza czy punkt nie leży na odcinku za pomocą metody Harris'a. Po tym etapie wyłonięte zostają punkty niezmiennicze. Algorytm przypisuje punktom orientację co powoduje niezmienniczość również względem orientacji. Można więc ustawić obrazy tak by odpowiadające sobie punkty charakterystyczne na każdym obrazie wskazywały jednakowy kierunek. Realizowane jest to poprzez wyznaczenie gradientu w punkcie charakterystycznym. Następnie wyznacza się gradienty w małym otoczeniu punktu charakterystycznego i z odpowiednio wyważonych gradientów tworzy się histogram. Z histogramu powstaje deskryptor który pozwala wyznaczyć orientację obszaru przetwarzanego w końcowym etapie. Etap końcowy polega na tworzeniu ostatecznych deskryptorów. Najczęściej przetwarzaniu podlega rozmyty wejściowy obraz. Etap przypomina ten poprzedni lecz otoczenie punktu charakterystycznego jest dużo większe i podzielone na cztery obszary. W każdym obszarze wyznacza się osobne deskryptory co kończy algorytm. Uzyskuje się w ten sposób punkty charakterystyczne wraz z dokładną orientacją.

Łączenie obrazów

Chcąc zakończyć tematykę tworzenia panoramy należało by wspomnieć w jaki sposób za pomocą metod klasycznych oraz tych bazujących na punktach kluczowych zostaje utworzony jednolity obraz panoramiczny. Nie trudno zauważyć, że w każdej z metod wykryte cechy pozwalają dopasować obrazy do siebie. Następnie nakłada się jeden obraz na drugi w taki sposób by występowała największa liczba pasujących elementów np. wykrytych linii, krawędzi, narożników czy deskryptorów. Następnie obrazy zostają scalone w jeden tzn. dokonuje się ich fuzji. Zazwyczaj polega to na przysłonięciu części obrazu drugim obrazem, lecz istnieją też podejścia, które uwzględniają, że na wspólnym obszarze informacje pozostają z obu zdjęć. Ostatnie podejście może spowodować pogorszenie się obrazu lecz strata informacji jest mniejsza.

Segmentacja obrazów

Segmentacja obrazów polega na znalezieniu obszaru, w którym występuje poszukiwany obiekt. W przypadku systemu TSR segmentacja obrazów polega na znalezieniu na obrazie znaku drogowego. By takie wykrywanie mogło mieć miejsce należy wyznaczyć zbiór cech tworzący jakościowy i ilościowy opis obiektu (znaku drogowego). Taki opis nazywany jest wzorcem. Zapisuje się go za pomocą wektora cech: $x = [x_1, x_2, \dots, x_n]$. Dla ułatwienia opisu obiektów zostają również wprowadzone klasy wzorców, zawierające wzorce z podobnymi wektorami cech. Klasę wzorców oznacza się za pomocą $\omega_1, \omega_2, \dots, \omega_M$ (M - numer klasy). Rozpoznanie wzorców nazwane klasyfikacją polega na przyporządkowaniu wzorców do ich klas. Wyznaczenie wzorców powinno się opierać o starannie dobrane cechy. Cechy takie

powinny mieć specyficzne własności. Ważne jest by wybrane cechy przypisane do obiektu przyjmowały różne wartości dla różnych klas obiektów. Ważnym atrybutem każdej cechy powinna być jej niezawodność. Niezawodność powinna polegać na przyjmowaniu podobnych wartości dla wszystkich obiektów danej klasy. Dodatkowo cechy wybrane wzorca powinny być nieskorelowane ze sobą. Skorelowanie można opisać za pomocą współczynnika korelacji:

$$\sigma_{x,y} = \frac{\frac{1}{P} \sum_{i=1}^P (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

gdzie P oznacza liczbę klasyfikowanych obiektów, μ wartości średnie, a σ odchylenie standardowe zbioru cech. Jeżeli współczynnik korelacji $\sigma_{x,y} = 1$ (-1) To cechy x i y uważa się za silnie skorelowane i jedną z nich można odrzucić. Ważne jest by dla obiektu nie przyjmować zbyt dużej ilości cech, ponieważ złożoność obliczeniowa detektorów obiektów rośnie wraz z ilością cech. Wykrywanie obiektów na obrazach może się odbywać na wiele sposobów. Kilka metod wykrywania i rozpoznawania obiektu zostały przedstawione poniżej. Wpierw zostało omówione podejście na podstawie koloru, a następnie kształtu. Zazwyczaj metoda wykrywania obiektu jest ściśle połączona z metodą rozpoznawania obiektu. Niekiedy używa się połączenia metod opartych na różnych podejściach. Oznacza to, że wpierw zostaje wykonane progowanie w określonym kolorze, a następnie rozpoznaje się kształty wyłonięne przez progowanie.

Wykrywanie na podstawie koloru

Wykrywanie obiektów na podstawie koloru stosuje się zazwyczaj w przypadku obiektu którego kształt jest skomplikowany. Czasem wykrywanie na podstawie koloru stosuje się dla obiektów których kształt jest dobrze określony ze względu na szybkość działania takiego algorytmu. Oczywiście ważnym aspektem jest to by kolor obiektu znacząco odróżniał się od kolorów tła. Określenie koloru może być zrealizowane za pomocą pobrania próbek z kilku wzorcowych obiektów. Takie czynność wykonywane jest zazwyczaj za pomocą ręcznego oznaczenia koloru do pobrania. Następnie wyznacza się średni kolor i dopuszczalny zakres odchyłki. Następnie na obrazach testowych wykonuje się progowanie w ustalonym kolorze. Mimo prostoty metody w praktyce takie wykrywanie obiektów jest zazwyczaj mało skuteczne, ponieważ kolor obiektu w dużej mierze zależy od oświetlenia. Możliwe jest polepszenie wykrywania obiektów wykrywając ich teksturę. Uwzględnia się wtedy regularność powierzchni zazwyczaj przez analizę częstotliwościową występowania zmian, które są charakterystyczne.

Zmiana przestrzeni barw (zredagować)

Przestrzeń barw RGB jest bardzo zmienna w zależności od warunków dlatego proponuje się przejście na przestrzeń barw HSI (ang. Hue, Saturation, Intensity). Pozwala to w pewnym stopniu zniwelować różne wahania koloru w zależności od pory dnia, rodzaju atmosfery czy innych zaburzeń świetlnych. Niestety przestrzeń HSI również nie jest wolna od wszelkich wad, np. nie uwzględnia temperatury barwy. Dodatkowo koszt obliczeniowy transformacji obrazu z RGB do HSI jest bardzo duży z powodu jej nieliniowości. Nieco skuteczniejszym podejściem jest przejście na przestrzeń LCH (ang. Lightness, Chroma, Hue) uzyskanej za pomocą modelu CIECAM97. Przestrzeń LCH uwzględnia temperaturę barw. Część autorów w swoich pracach jednak stwierdza, że zmiana przestrzeni barw jest zbędna ponieważ unormowana przestrzeń RGB jest wystarczająco dobra. Niektóre sieci neuronowe [10] o których wspomniano w późniejszych rozdziałach wykorzystują inną przestrzeń barw YUV. Przestrzeń posiada 3 kanały 1 kanał jasności i dwa kanały chrominancji i jest używana m. in. w odbiornikach telewizyjnych.

Warto wspomnieć o pracy [11] gdzie model został oparty na kolorach ale wykorzystano coś więcej niż samo progowanie. Zaproponowany model korzysta z kaskadowego klasyfikatora AdaBoost. AdaBoost oparty na boostingu algorytm zaprezentowany 1996 roku przez Yoav Freund i Robert Schapire [12]. Algorytm z wielu słabych klasyfikatorów tworzy jeden mocny klasyfikator. W przypadku wcześniej wspomnianej pracy [11] autorzy dodatkowo pomniejszyli rozdzielczości kolorów. Wszystkie odcienie kolorów występujących na znakach są rzutowane na ich podstawowe odpowiedniki. Tam gdzie skupiska kolorów odpowiadają tym na znakach (np. czerwony i biały) tam jest zaznaczany obszar z potencjalnym znakiem.

Model prawdopodobieństwa barwy

Model prawdopodobieństwa kolorów uzyskuje się na podstawie rozkładu kolorów znaków drogowych, które są szacowane na podstawie ręcznie pobranych próbek szkoleniowych. Aby poprawić odporność na zmiany oświetlenia, wartości RGB są konwertowane na przestrzeń Ohta [22], ponieważ najlepiej sprawdza się w naszych eksperymentach.

Założmy, że są $N-1$ kolory znaków drogowych, a wszystkie tła są oznaczone innym kolorem. Najpierw ręcznie zbieramy wartości RGB tych N kolorów z obrazów treningowych. Następnie przekształcamy te wartości RGB w przestrzeń Ohta według

$$P_1 = \frac{1}{\sqrt{2}} \frac{R - G}{R + G + B}$$

$$P_2 = \frac{1}{\sqrt{6}} \frac{G - R - B}{R + G + B}$$

Pełny opis metody można znaleźć w [13].

Uzyskane mapy prawdopodobieństwa są szarymi obrazami, w których wysokie intensywności wskazują na obecność określonych kolorów. Pierwszy rząd z rys. 2 pokazuje przykład map prawdopodobieństwa. Rys. 2 (a) jest wejściowym obrazem kolorowym, rys. 2 (b) i (c) są odpowiednimi mapami prawdopodobieństwa odpowiednio czerwonego i niebieskiego. Jak można zauważyć, czerwone piksele na oryginalnym obrazie mają duże natężenie na rys. 2 (b). Podobnie niebieskie piksele mają duże natężenie na rys. 2 (c). Mapy prawdopodobieństwa zwiększają kontrast między znakami drogowymi a tłem, dzięki czemu wykrywanie znaków drogowych jest o wiele łatwiejsze.

Aby nasz model prawdopodobieństwa kolorów był dostępny dla aplikacji czasu rzeczywistego, wstępnie obliczamy tabelę wyszukiwania (LUT), aby przyspieszyć obliczenia. Podczas wykrywania online obliczamy po prostu indeks każdego piksela według jego wartości RGB i znajdujemy odpowiadające mu prawdopodobieństwo w LUT. Przy pomocy LUT czas na obliczenie map prawdopodobieństwa dla 1360×800 obraz może zostać zredukowany z kilku minut do około 30 ms na normalnym komputerze (czterordzeniowy procesor Intel 3,4 GHz, 4 G RAM)

Wykrywanie na podstawie kształtu

Równie powszechnym podejściem jak wykrywanie obiektów na podstawie koloru jest wykrywanie obiektów na podstawie kształtu. W przypadku znaków drogowych kształt znaku jest konkretnie określony. Znaki informacyjne są kwadratowe, nakazu lub zakazu okrągłe, a ostrzegawcze trójkątne. Wykrywanie kształtów jest problematyczne ze względu na zmienność w kształcie znaku w zależności z jakiego kąta znak jest obserwowany. Znak może być

częściowo zasłonięty co również wprowadza komplikację. Wykrywanie figur geometrycznych na obrazie może być zrealizowane na kilka sposobów. Te ważniejsze zostały przedstawione poniżej.

Wykrywanie krawędzi

Wykrywanie krawędzi jest najbardziej popularną metodą stosowaną w uwidacznianiu obiektów na obrazie. W poprzednim rozdziale opisywano wykrycie krawędzi przez algorytmy klasyczne na potrzeby łączenia obrazów w panoramę. Nic nie stoi na przeszkodzie by raz wykryte krawędzie stosować do wykrywania obiektów. Najczęściej krawędź identyfikowana jest jako przejście z obszaru ciemniejszego do jaśniejszego bądź na odwrót. Można stwierdzić, że jest to granica pomiędzy dwoma obszarami o różnych jasnościach. Takie podejście wymaga ustalenia progu zmiany jasności jaki powoduje wykrycie krawędzi. Większość metod bazuje na operatorach gradientowych ustalając lokalny, zamiast globalnego próg wykrywania krawędzi. Metody te wykorzystują zmiany pierwszej lub drugiej pochodnej obrazu w skali szarości. Przykładem takiego algorytmu wykrywania krawędzi jest Operator Robertsa sprawdzający różnicę między sąsiednimi pikselami w obrazie. Niestety jest on bardzo czuły na zakłócenia w postaci szumów. W większości przypadków do wykrywania krawędzi stosuje się operator Canny lub innej bardzo podobnej metody. Operatora Sobela czyli splotu macierzy z obrazem [14].

Detektor Canny

Metoda zaprezentowana w Johna F. Canny w 1986 roku jest jedną z bardziej popularnych metod wykrywania krawędzi. Metoda z powodzeniem wykrywa krawędzie w pewnym stopniu odfiltrowując miejsca gdzie krawędzie nie powinny być wykryte. W pierwszym kroku następuje filtracja obrazu filtrem Gaussa. Powoduje to wstępnie odfiltrowanie zakłóceń. Następnie za pomocą np. operatora Sobela wykrywa się gradient zmian poziomych G_x i pionowych G_y każdego punktu na obrazie. Następnie wyznacza się długość $G = \sqrt{G_x^2 + G_y^2}$ i kąt detekcji krawędzi $\theta = \arctan(\frac{G_y}{G_x})$. Niekiedy kąt zaokrągla się do wartości liczonych co 45° . Następnie odrzuca wykryte piksele, które nie łączą się z żadną pobliską krawędzią i dołącza piksele na zakończeniach wykrytych krawędzi jeżeli próg wykrywania je odrzucił. Na obrazie z wykrytymi krawędziami można starać się lokalizować obiekty.

Wykrywanie rogów

Tak jak metody klasyczne łączą się z wykrywaniem obiektów na podstawie krawędzi tak metody wykrywania narożników jako punktów kluczowych łączą się z metodami detekcji obiektów na podstawie rogów. Narożniki wykryte np. detektorem Harris'a w poprzednim etapie mogą posłużyć do wykrywania obiektów w obecnym. Głównie przydaje się to w znakach ostrzegawczych i informacyjnych.

Transformata Hougha

Metodą na którą należało by zwrócić uwagę zastosowanie filtru Hougha, która pozwala odnajdywać regularne kształty. Została ona zaprezentowana w 1962 roku jako metoda wykrywania linii prostych ale udało się ją zastosować również dla wykrywania regularnych kształtów takich jak okręgi. W 1993 roku Anagnou przestawił udoskonaloną metodę pozwalającą uzyskanie większej rozdzielczości kątowej oraz radialnej znajdowanej linii. Omówienie metody zostało zrealizowane na przykładzie wykrywania kształtu kwadratowego. W tej metodzie każda prosta opisana jest równaniem $\rho = x \cos \theta + y \sin \theta$. Początkowo

zostaje zaimplementowana tablica (komórki nazywa się akumulatorami) z wartościami zerowymi. Następnie każdy punkt obrazu przekształcany jest w dyskretną krzywą sinusoidalną w przestrzeni $Op\Theta$. Oblicza się wartości dla $\Theta=(0,360^\circ)$. Obliczenie wartości ρ uzyskuje się przez dodanie jedynek w odpowiednie pola tablicy. Po wykonaniu inkrementacji dla wszystkich punktów komórki z największą wartością ρ . Położenie komórki określają parametry prostej zaś wielkość ρ określa długość odcinka. Jeżeli wykryte proste mają podobną długość i przecinają się pod kątem prostym to można uznać, że kształt jest kwadratowy. Transformacja Hougha odnosi sukcesy w wykrywaniu linii czy okręgów lecz do wykrycia innych kształtów jak np. trójkąty wymaga skomplikowanych obliczeń i jest obciążająca pamięciowo.

Oryginalna metoda Hougha służy do wykrywania prostych. Metodę tę później uogólniono na wykrywanie kształtów dających się opisać analitycznie np. okręgów (Richard Duda and Peter Hart, 1972)[3] oraz na wykrywanie dowolnych kształtów (Dana H. Ballard, 1981)

Wykrywanie na podstawie tekstury

Wykrywanie obiektów na podstawie koloru i kształtu nie zawsze daje dobre wyniki, zaczęto szukać innych rozwiązań mogących poprawić skuteczność wykrywania. Jak wiadomo tekstura to graficzny odpowiednik faktury. Faktura obiektów w rzeczywistości bardzo wpływa na postrzeganie obiektów przez człowieka dlatego naukowcy od razu zaczęli zgłębiać zagadnienie dla rozpoznawania obiektów. Kilka metod wykrywania obiektów na podstawie tekstury zostało przedstawione poniżej.

Filtry Gabora

[here] opisać

HOG

HOG (ang. Histograms of Oriented Gradients) - histogram gradientów zorientowanych jest deskryptorem obrazu pozwalającym na znalezienie kształtu i wyłonienie obiektu. Idea algorytmu polega na podzieleniu obrazu przekonwertowanego do skali szarości na małe fragmenty (np. 8x8 px.) i obliczeniu dla każdego piksela w fragmencie różnicy jasności pomiędzy pikselami sąsiadującymi (gradientu). Następnie dla fragmentu wyznacza się histogram gradientu. W przeciwieństwie do wcześniej przedstawionego algorytmu SIFT, HOG oblicza deskryptory w równomiernie określonych fragmentach obrazu. Zastosowanie lokalnej normalizacji kontrastu w nakładających się na siebie regionach poprawia skuteczność wykrywania obiektów

Zaproponowany w pracy [15] wariant HOG (HOGv) posiada dwie modyfikacje poprawiające wykrywanie obiektów. Pierwszą z nich jest uwzględnienie zarówno wrażliwych jak i niewrażliwych na kontrast orientacji gradientów, w taki sposób, że bardziej szczegółowe lokalne informacje o znakach mogą być włączone do zgromadzonych histogramów. Drugą modyfikacją jest to, że każdą komórkę normalizuje się odpowiednio z czterema sąsiednimi blokami. Znormalizowane histogramy komórki są wymiarowo zmniejszane za pomocą strategii podstawowej analizy komponentów (PCA). Takie działanie ma na celu usunięcie nadmiarowych informacji.

LBP

LPB (ang. local binary patterns) zostało opisane w 1994 roku [16] jako algorytm do wykrywania obiektów. Algorytm jest bardzo podobny do algorytmu HOG. Na potrzeby LBP ramka obrazu zostaje podzielona na mniejsze obszary. W każdym obszarze piksele porównuje

się w ustalonej kolejności z pikselami sąsiadującymi i w zależności czy jego wartość jest większa lub równa czy mniejsza wpisuje się odpowiednio 1 lub 0. Dzięki takiemu rozwiązaniu gdy zostaje zmieniona jasność obrazu, skuteczność algorytmu jest taka sama. W każdej fragmencie z wyznaczonych wartości tworzy się histogram. Wartości na osi x przedstawione są co 45° . Następnie histogram zostaje unormowany i zostaje ustalony gradient całego okna. Po stworzeniu gradientów w każdym oknie następuje połączenie podobnych gradientów ze sobą i próba rozpoznania obiektu. Jeżeli wykryte gradienty są podobne do tych, które posiada obiekt jaki chciano wykryć algorytm uznaje, że obiekt został wykryty. Algorytm odniósł duży sukces w wykrywaniu twarzy dlatego postarano się go użyć do wykrywania innych obiektów. Ponieważ algorytm do swojego działania wykorzystuje dużą ilość obliczeń można wprowadzić obszar zainteresowań ROI (ang. Region of Interest), który zawęży strefę obiektu. Takie podejście zaproponowano w pracy [17]. Rozwiązanie polega na ustaleniu potencjalnych obszarów poszukiwań za pomocą okna przesuwanego o niewielkich rozmiarach. Etap nazywa się filtrowaniem zgrubnym i używa algorytmu LDA. Następnie podczas dokładnego filtrowania weryfikuje się okna odnalezione w etapie poprzednim oraz za pomocą metody NMS (ang. non-maximal suppression) wykonywane jest filtrowanie okien mogących odnosić się do tego samego obszaru. Klasyfikacja obiektu wykonywana jest za pomocą maszyny wektorów nośnych, która została opisana w następnym rozdziale.

ACF

ACF łączy informacje o kolorach i gradientach, co jest lepsze od HOG [18] Integral channel features [here]

Podejście hybrydowe

Podejście hybrydowe polega na połączeniu metod opartych na kolorze, kształcie i fakturze. Zazwyczaj w początkowym etapie następuje progowanie w określonych kolorach co zawęży obszar poszukiwań. Następnie w odnalezionych regionach wykorzystuje się podejścia oparte na kształcie i fakturze w celu poprawy wydajności wykrywania.

HOG i SVM

Maszyna wektorów nośnych SVM (ang. Support Vector Machine) jest klasyfikatorem umożliwiającym określenie do jakiego zbioru należy obiekt. Maszyna do poprawnego działania potrzebuje zbioru uczącego, w którym przedstawione są poprawnie sklasyfikowane obiekty. SVM znajduje hiperpłaszczyznę (funkcję) oddzielającą dwie klasy. Jeżeli taka funkcja nie istnieje w przestrzeni zadania należy wprowadzić funkcję jądra, która powiększy przestrzeń. System przedstawiający rozwiązanie dla problemu TSR został przedstawiony w pracy [19]. Działanie systemu polega na wykrywaniu obszary występowania znaków za pomocą progowania w przestrzeni HSI. Znaki białe wykrywane są za pomocą rozkładu achromatycznego. Każdy obszar prawdopodobnego wystąpienia znaku wpisywany jest w kwadrat. Następnie wyznacza się 20 odległości liczonych od ramki kwadratu do znalezionej koloru w środku ramki mający być obwiednią obiektu. Z tych 20 odległości tworzy się wektor DtB dostarczany jest na wejście SVM. Warto wspomnieć, że wyznaczony w ten sposób wektor znaków ośmiokątnych niezbyt różni się od znaków okrągłych dlatego znaki ośmiokątne klasyfikowane są dopiero na etapie rozpoznawania znaku. Następnie przy użyciu kilku wytrenowanych maszyn SVM następuje klasyfikacja obszaru. Każda maszyna potrafi rozpoznać jeden typ obrazu. W zależności od wykrytego koloru (czerwonego, niebieskiego, żółtego lub białego) algorytm wyznacza inne maszyny do klasyfikacji. Proces rozpoznania

opiera się na maszynie SVN z ziarnami Gaussa. Dużą zaletą metody jest jej odporność na rotację.

Maszyna wektorów nośnych może być również połączona z algorytmem tworzącym histogram gradientów zorientowanych. Takie podejście zaprezentowane zostało w pracy [13]. Wykrywanie obiektów za pomocą metody HOG wspomagane jest informacją o kolorze, co jest ogromnie istotne dla wykrywania znaków drogowych. Oryginalnie algorytm HOG oblicza gradienty dla każdego kanału kolorów i przyjmuje gradient o największej normie [20], zaś algorytm w opisywanej pracy [13] oblicza funkcje HOG dla każdego kanału kolorów po czym łączy je ze sobą tworząc histogram. W odróżnieniu od podobnych metod, obliczenie funkcji HOG następuje na podstawie mapy prawdopodobieństwa, ponieważ mapa prawdopodobieństwa może kodować informacje o kolorze i kształcie znaku drogowego, jednocześnie tłumiąc wpływ tła. Jednak na podstawie samej mapy prawdopodobieństwa nie ma możliwości zidentyfikowania znaku. Aby rozwiązać ten problem, dodano dodatkową funkcję w postaci klasycznego algorytmu HOG. Do identyfikacji przeszkolony został wielopoziomowy klasyfikator SVM. Autorzy algorytmu przetestowali swoją pracę na zbiorze danych GTSDb, gdzie istnieją trzy kategorie znaków drogowych, dlatego wyszkolony został 4-klasowy klasyfikator SVM z dodatkową klasą tła. Za jądro klasyfikatora SVM została wybrana funkcja radialna, ponieważ najlepiej sprawdziła się w eksperymentach. Zaproponowany algorytm na bazie danych GTSDb wykrywa znaki zakazu i nakazu z 100 % poprawnością, znaki ostrzegawcze z 94.29 % poprawnością. Czas wykrycia znaków w ramce to 0,067 sekundy co jest bardzo zadowalającym wynikiem.

SVF

SVF (ang. Simple Vector Filter) jest filtrem zaproponowanym w 2002 roku przez autorów artykułu [21] pozwalającym wyłonić obiekty z obrazu na podstawie informacji o kolorze w przestrzeni barw HSI. Jak można zobaczyć w artykule w połączeniu z filtrem krawędziowym pozwala dobrze usunąć tło pozostawiając same znaki. Do identyfikacji autorzy proponują metodę bazującą na algorytmie genetycznym.

Inne podejścia

Algorytm Viola i Jonesa

Algorytm Viola–Jones zaproponowany w 2001 przez Paul Viola i Michael Jones wprowadzie został stworzony do wykrywania twarzy [22], lecz znakomicie nadaje się do wykrywania obiektów każdego typu na obrazie. Takie podejście zostało przedstawione w pracy [1]. W pracy jako klasyfikatora używa się algorytmu AdaBoost (ang. Adaptive Boosting). Algorytm AdaBoost jest oparty na boostingu. Tworzy on z kilku mniejszych klasyfikatorów jeden silny klasyfikator. Dla ww. pracy jako słabe klasyfikatory przyjmuje się drzewa decyzyjne, w których każdy węzeł jest jedną z cech haara-podobnych. Cechy Haara swoją nazwę zawdzięczają falką Haara zaproponowanym przez Alfréda Haara w 1909 lub 1910 roku. Cechy Haara to najprościej mówiąc jest to pewna maska, która posiada dwa typy pikseli. Po jej nałożeniu na część obrazu sumuje się jasności pikseli należących do danego typu. Jeżeli różnica między dwoma typami jest wystarczająca cecha Haara daje pozytywną odpowiedź. Algorytm używa niewielkiej ramki przesuwanej po obrazie. W każdym położeniu ramki za pomocą cech Haara stara się odnaleźć miejsce występowania znaku. Cechy Haara sprawdzane są kaskadowo co pozwala od razu odrzucić rejony gdzie znak nie występuje. By przyspieszyć obraz zostaje scałkowany. W wcześniej już wspomnianej pracy [1] metoda pozwoliła na wykrywanie nie tylko znaków drogowych lecz również na wykrycie samochodów a nawet rowerzystów.

Ponieważ algorytm oprócz wykrywania znaków wykrywa dwa inne typy obiektów, wprowadzono dodatkowo funkcje subkategoryzacji. Wielką zaletą metody jest wykrywanie obiektów w czasie rzeczywistym.

Praca, która również korzysta z metody opartej na AdaBoost i cechach Haara została zaprezentowana w [23]. Dzieło pozwala na wykrywanie i śledzenie obiektów. Cechy Haara zostały również wsparte informacją o kolorach co pozwoliło na redukcję błędów pierwszej klasy z 1.6 % do 1.4 %, oraz błędów drugiej klasy z 0.3 % do 0.03 %, co jest spektakularnym wynikiem.

Mapa cieplna

Wartym wspomnienia podejściem detekcji znaków jest tworzenie mapy cieplnej obszarów występowania znaków. Taka mapa działa na zasadzie algorytmów biologicznych. Autorzy artykułu [24]. [here] (opisać artykuł)

Symetria promieniowa

Niemniej jednak, niektóre ostatnie prace (patrz [25] i [26]) wdrożyły szybki algorytm oparty na symetrii promieniowej, która jest dostosowana do kształtów trójkątnych, kwadratowych, diamentowych, ośmiokątnych i okrągłych. Działa na gradientach obrazu w skali szarości i wykorzystuje naturę kształtów, które głośnią w punkcie środkowym dla okrągłych znaków i linii głośń w przypadku regularnych wielokątów. Główną zaletą tej metody jest to, że jest ona w stanie działać w czasie rzeczywistym. Ponieważ wykrywa kształty oparte na krawędziach, algorytm jest odporny na zmiany oświetlenia.

Wnioski

Podsumowując zagadnienie wykrywania znaków drogowych na obrazie należy stwierdzić, że na dzień dzisiejszy najpowszechniejsze są algorytmy hybrydowe. Podejścia na kolorze choć praktycznie bezużyteczne jako odosobniony sposób znajdowania znaków wspaniale sprawdzają się jako wspomaganie dla wykrywania na podstawie kształtu [21] czy tekstury. Podejścia na podstawie kształtu sprawdzają się dużo lepiej od podejść na podstawie koloru, ponieważ są nie wrażliwe na światło. Niestety są także bardziej kosztowne obliczeniowo. Podejścia oparte na teksturze używają najczęściej bazują na gradientach HOG, LPB, ACF. Takie podejście jest pozwala z dużą dokładnością wyłonić obrys znaku drogowego, co na etapie identyfikowania może posłużyć do szkolenia klasyfikatora. Najszybsze a zarazem najbardziej odporne są podejścia hybrydowe. Niektóre podejścia [19], [13] wykorzystują funkcje HOG z SVM, inne [28] używają funkcji ACF z klasyfikatorem AdaBoost. Oprócz powyższych podejść, spłotowa sieć neuronowa (CNN) jest stosowana do wykrywania znaków drogowych i osiąga doskonałe wyniki [29].

Klasyfikacja znaków drogowych

Klasyfikowanie znaków drogowych odnosi się do rozpoznania jaki znak został odnaleziony przez segmentację. Zazwyczaj znak, który należy zidentyfikować wypełnia znaczny obszar obrazu, ponieważ segmentacja zawęziła obszar przekazując do identyfikacji tylko najistotniejszą część. Obecnie najpowszechniej stosowanym klasyfikatorem obrazów są sieci neuronowe. Już w artykule z 1996 roku [14] używa się sieci do wykrycia typu znaku drogowego. Poniżej zostaną opisane typy sieci używane na potrzeby TSR. Warto jednak wspomnieć o innych algorytmach. Niekiedy do rozpoznawania typu znaku stosuje się algorytmy genetyczne, które również zostaną przedstawione poniżej.

Sieci neuronowe

[here](opis sieci) Oczywiście każda sieć na wejściu posiada określoną liczbę neuronów dlatego należy obraz wejściowy przeskalować, znormalizować do tej liczby. Do normalizacji obrazu można stosować wiele metod. Najprostszą z nich jest odrzucenie części pikseli. Możliwa jest również interpolacja dwuliniowa. Sieć neuronowa do klasyfikacji.

Perceptron wielowarstwowy

Perceptron wielowarstwowy jest siecią neuronową wykrywającą znaki drogowe zaprezentowaną we wcześniej wspomnianej pracy [14]. Autorzy użyli dwóch sieci. Jednej dla znaków trójkątnych, a drugiej dla okrągłych. Wielkość wejścia sieci odpowiadała obrazowi o rozmiarach 30x30 piksel, dlatego każdy otrzymany obszar przez segmentację należało znormalizować do takiego rozmiaru. Warstwa wyjściowa sieci posiadała 10 wyjść z czego 9 wyjść odpowiadało znakom. Ostatnie wyjście informowało, że znak nie został rozpoznany. Autorzy eksperymentalnie dobrali wielkość sieci z pośród 3 typów sieci. Najlepsze wyniki uzyskała 3 konfiguracja 30x30/15/5/10 z 98% skutecznością dla najlepiej rozpoznawalnych znaków. Rozpoznawanie na jednostce obliczeniowej PC486 33 MHz zajmuje 220 ms dla obrazu w rozdzielczości 256x256 co jest znakomitą wynikiem jak na rok 1996. Należy jednak zaznaczyć, że segmentacja w tej pracy była bardzo kosztowna czasowo, co nie pozwoliło algorytmowi działać w czasie rzeczywistym.

NN

Sieć neuronowa wstecznej propagacji (ang. Backpropagation Neural Network) do rozróżniania typu znaku drogowego została zaprezentowana w pracy [27]. W tej pracy detekcja obszaru występowania znaku przeprowadzona jest za pomocą progowania oraz algorytmów genetycznych. Sieć uczona została według paradygmatu ART1 (ang. Adaptive Resonance Theory). Posiada dwie warstwy ukryte. Po nauczaniu sieci rozpoznanie obrazu zajmuje 8.8 sekundy na jednostce AMD Duron 1 GHz.

Sieć neuronowa propagacji wstecznej została również zaimplementowana przez autorów artykułu [30]. Autorzy wspomnieli, że jest kosztowny pod względem obliczeniowym do szkolenia i łatwo wpasowuje się w lokalne optimum.

W pracy [31] opracowano algorytm bazujący na sieci neuronowej propagacji wstecznej pozwalający na śledzenie obiektów. Dwie sieci działają równolegle ze sobą. Jedna stara się zidentyfikować obiekt na bazie koloru, a druga na bazie wykrytych krawędzi. Obiekt jest rozpoznawany i klasyfikowany a następnie śledzony.

DNN

Głęboka sieć neuronowa jest to sieć składająca się z wielu warstw ukrytych. Mechanizm uczenia się DNN nie może zagwarantować odporności na zaszumienie obrazu wejściowego np. przez obrót i skalowanie, chyba, że niektóre próbki szkoleniowe były zaszumione. Co więcej, ich koszt obliczeniowy podczas szkoleń i procesów rozpoznawania jest kosztowny.

Podejście opierające się na głębokich sieciach prezentuje artykuł [32]. Do testów został użyta jednostka Intel Xeon E5-1620, dwoma kartami NVIDIA Tesla K20 G-PU i 32 GB pamięci. W artykule zostały zaproponowane dwie sieci neuronowe. Jedna z nich przystosowana była do detekcji, a druga do detekcji i klasyfikacji.

CNN

CNN (ang. convolutional neural network) splotowe sieci neuronowe są sieciami, których pierwsze warstwy dokonują pewnego splotu danych wejściowych. W przypadku przetwarzania obrazów splot może polegać na łączeniu informacji z kilku pikseli i przekazywaniu jej jako ujednoliconą wartość. W pracy [13] zastosowano splotową sieć neuronową dla klasyfikacji znaku drogowego. W tym celu szkolono trzy sieci odpowiadające, każda innemu typowi znaku. Na wejście sieci dostarcza się obraz w skali szarości o rozmiarze 32×32 pikseli. Ponieważ obrazy są przechwytywane w różnych warunkach oświetleniowych i pogodowych, znaki tej samej klasy mogą znacząco różnić się od siebie. Aby zmniejszyć różnicę użyto metody korelacji adaptacyjnego histogramu ograniczonego kontrastem (CLAHE), pozwalającej dostosować kontrast obrazów. Ponieważ wykryte regiony mogą zawierać fałszywe znaki przedstawiające tło, dodano dodatkową klasę tła w trakcie uczenia. Każda sieć CNN zawiera dwie warstwy splotowe i dwie warstwy próbkowania, a także pełne połączenie MLP na dwóch ostatnich warstwach. Rozmiar jądra filtra w obu warstwach splotowych wynosił 5×5 . Ponieważ rozmiar obrazu wejściowego wynosi 32×32 , po pierwszej warstwie splotu, zostało dodanych się 16 map cech powodując zmniejszenie rozmiaru wejścia drugiej warstwy splotowej najpierw do rozdzielczości 28×28 , a następnie warstwa pod próbkowania zmniejsza wejście do 14×14 . Po drugiej warstwie pod próbkowania, 32 mapy funkcji zmieniają na rozmiar 5×5 . Następnie te mapy funkcji są przekształcane na długi wektor o długości 800 znaków. Algorytm jest zaimplementowany w języku C++ z włączoną opcją OPENMP. Wykorzystywane są również implementacje MSER, HOG i SVM z biblioteki OpenCV. Wyszkolono sieć neuronową za pomocą Torch7 i przepisano wyszkolone wartości do kodu C++. Algorytm został uruchomiony na komputerze z czterordzeniowym procesorem 3,7 GHz.

Autorzy artykułu [33] zaproponowali pewne ulepszenie dla sieci. Sieć posiada 6 warstw ukrytych. Ulepszenie polega na dodaniu klasyfikatora po pierwszej warstwie splotowej. Oczywiście sieć również przekazuje pełną informację do warstwy trzeciej. Takie rozwiązanie spowodowało, że metoda na zbiorze danych GBTSR osiągnęła dokładność wykrywania na poziomie 99,7 % co jest jednym z lepszych wyników.

W pracy [34] zauważyli, że sieci splotowe są wydajniejsze, gdy są używane w przesuwanym oknie, ponieważ wiele obliczeń może być ponownie wykorzystanych w nakładających się regionach.

Splotowe sieci neuronowe użyte były również w pracach [35], [36], [37], [38], [39], [40], [41], [42]

ELM

ELM (ang. Extreme Learning Machine) jest algorytmem uczenia się pojedynczych sieci ukrytych sieci neuronowej (SFNN). Pierwszą zaletą algorytmu ELM jest to, że wartości wejściowe między wejściem i warstwami ukrytymi są losowo przypisywane, co pozwala losowe mapowanie cech. Ponieważ wyszkolone są tylko warstwy wyjściowe między warstwami ukrytymi i wyjściowymi, nie jest wymagane strojenie za pomocą wstecznej propagacji warstwa po warstwie. Algorytm ELM może uzyskać optymalne i uogólnione rozwiązanie do rozpoznawania wieloklasowego. Dodatkowo, łatwo można rozszerzyć ELM na sieć wielowarstwową lub głęboką za pomocą techniki autoencoder. Metoda ELM została również wykorzystana do modelowania lokalnych pól recepcyjnych i wykorzystywana do

uczenia na dużych zbiorach danych. W związku z tym zastosowanie ELM dla TSR może dać lepsze rozwiązanie. Co więcej, z powodu losowego przypisania wag wejściowych, algorytm ELM może zmniejszyć koszt obliczeniowy szkolenia. Ponieważ istnieje tylko jedna ukryta warstwa, szybkość obliczeniowa procesu rozpoznawania jest również szybka.

Praca [15] stosuje jako klasyfikator pojedynczą sieć z ukrytą warstwą. W oparciu o algorytm ELM, połączenie pomiędzy warstwami wejściową i ukrytą realizuje mapowanie cech losowych, podczas gdy tylko wagi pomiędzy warstwami ukrytymi i wyjściowymi są wyszkolone. W rezultacie strojenie warstwa po warstwie nie jest wymagane. Tymczasem norma wag wyjściowych jest zawarta w funkcji kosztów. W związku z tym, Klasyfikator oparty na ELM może osiągnąć optymalne i uogólnione rozwiązanie dla wieloklatkowych TSR. Ponadto może zrównoważyć dokładność rozpoznawania i koszty obliczeniowe.

LDA

LDA (ang. Linear Discriminant Analysis) - liniowa analiza dyskryminacyjna jest to algorytm, którego używa się w uczeniu maszynowym do znalezienia liniowej kombinacji cech, które najlepiej rozróżniają dwie lub więcej klas obiektów lub zdarzeń. Wynikowe kombinacje są używane jako klasyfikator liniowy lub, częściej, służą redukcji wymiarów do późniejszej klasyfikacji statystycznej. [here] (usunąć cały opis??)

SVM

Część systemów TSR do klasyfikacji znaków drogowych korzysta z maszyny wektorów nośnych SVM (z ang. support vector machine). Przykładem takiego systemu jest wcześniej omawiany system z 2013 roku przedstawiony w pracy [17]. W przypadku pracy autorzy wyuczyli algorytm bazujący na maszynie wektorów do klasyfikacji znaków ostrzegawczych i informacyjnych. W przypadku tej pracy algorytm był szkolony za pomocą 600 różnych danych testowych pochodzących z zbioru GTSBD oraz testowany na 300 innych obrazach. Osiągnął dobre wyniki wykrywania niestety nie pozwalające pracować w czasie rzeczywistym. Implementacja została wykonana za pomocą programu Matlab i na jednostce Core i3 3.3 GHz. Możliwe, że po implementacji w języku C/C++ i na bardziej wydajnym procesorze szybkość wzrosła by na tyle, że można by było uznać że algorytm pracuje w czasie rzeczywistym.

We wcześniej już wspomnianej pracy [19] również używa się maszyny wektorów nośnych do rozpoznania obiektów. Rozpoznawanie realizowane jest przez maszyny SVM z jądrami Gaussa. Praca do treningu wykorzystuje bibliotekę LIBSVM. Starano się użyć funkcji liniowej jako hiperpłaszczyzny lecz nie zawsze było to możliwe. W trudnych przypadkach wprowadzono jądro Gaussa. Na wejście SVM podawano 31x31 pikseli dlatego każdy obszar gdzie zostały wykryte potencjalne znaki musiał być przeskalowany do tej rozdzielczości. Dla każdego koloru i kształtu podano od 20 do 100 próbek uczących. Autorzy algorytmu przekonują, że ich dzieło wykryło wszystkie znaki podczas jazdy z normalną prędkością zarówno w dzień jak i w nocy. Trzeba jednak wspomnieć, że podczas testu warunki pogodowe były idealne, a znaki nie były w żaden sposób zniekształcone.

GA

Drugim najbardziej popularnym podejściem po sieciach neuronowych do klasyfikacji są algorytmy genetyczne GA (ang. Genetic Algorithm). [here] (krótki opis). Jedną z wcześniej opisywanych prac wykorzystującą taki algorytm została przedstawiona w artykule [21]. Segmentacja w tej pracy odbywa się za pomocą wykrywania krawędzi i algorytmu SVF. Z algorytmu genetycznego również korzysta praca [27], która została przedstawiona wcześniej.

Warto wspomnieć też o pracy [43], która swoją premierę miała w roku 1996 i przedstawia podejście które łączy algorytm genetyczny i sieć neuronową.

K-d Tree

Drzewo wielowymiarowe Kd (ang. k-dimensional tree) jest drzewem binarnym. W pracy [44] użyto drzewa binarnego do wykrywania obiektów na obrazach poddanych transformacji HOG. Wykazało porównywalną wydajność z innymi najnowocześniejszymi metodami w zakresie szybkości obliczeniowej procesu rozpoznawania, ale jego dokładność rozpoznawania rośnie wraz z kosztem obliczeniowym, dlatego nie nadają się tak dobrze do zagadnienia TSR.

Podsumowanie

Najpowszechniej stosowanym podejściem do identyfikacji obiektów są splotowe sieci neuronowe. Algorytmy genetyczne zajmują drugie miejsce w powszechnym stosowaniu i mogą być użyte zarówno na etapie klasyfikacji jak i segmentacji. Drzewa wielowymiarowe nie nadają się do rozpoznawania znaków ponieważ są zbyt wolne.

Program

Propozycja rozwinięcia/konstrukcji metody

Przygotowanie środowiska badawczego i plan badań

Podsumowanie

Bibliografia

- [1] S. P. C. S. A. v. d. H. a. F. P. Q. Hu, „Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1002-1024, kwiecień 2016.
- [2] M. M. T. a. T. B. M. A. Mogelmose, „Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey,” *Transactions on Intelligent Transportation Systems*, pp. 1484-1497, grudzień 2012.
- [3] E. Lee, S.-S. Lee, Y. Hwang i S.-J. Jang, „Hardware implementation of fast traffic sign recognition for intelligent vehicle system,” w *2016 International SoC Design Conference (ISOCC)*, Jeju, 23-26 październik 2016.
- [4] C. A. V. D. M. G. S. P. Maes F., „Multimodality image registration by maximization of mutual information,” *Transactions on Medical Imaging*, p. 187–198, 1997.
- [5] F. J. Zitová B., „Image registration methods, a survey. Image and Vision Computing,” p. 977–1000, 2003.
- [6] T. Szymczyk, „Metoda dopasowania wzorców w rozpoznawaniu obrazów – ograniczenia, problemy i modyfikacje,” w *Automatyka*, 2008, pp. Tom12, Zeszyt 2.

- [7] B. P. M. B. Marcin Kondej, „Szybki algorytm dopasowania obrazów dla potrzeb fuzji w czasie rzeczywistym,” *Projekt rozwojowy MNiSW nr. O R00 0019 07*, 25 listopad 2010.
- [8] S. M. Heather J.P., „New adaptive algorithms for real-time registration and fusion of multimodal imagery,” w *Proc. SPIE*, 2010.
- [9] D. Lowe, „Scale-invariant feature transform,” *University of British Columbia*, 1999.
- [10] Y. L. Pierre Sermanet, „Traffic sign recognition with multi-scale Convolutional Networks,” w *The 2011 International Joint Conference on Neural Networks*, San Jose, 31 lipca - 5 sierpnia 2011.
- [11] M. S. K. D. I. I. H. M. D. Deguchi, „Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples,” *Proc. IEEE IV Symp.*, pp. 72-77, 2011.
- [12] Y. F. R. E. Schapire, „Experiments with a New Boosting Algorithm,” w *Machine Learning: Proceedings of the Thirteenth International Conference*, New Jersey, 199.
- [13] H. L. H. X. F. W. Y. Yang, „Towards Real-Time Traffic Sign Detection and Classification,” *IEEE Transactions on Intelligent Transportation Systems*, tom 17, nr 7, pp. 2022 - 2031, czerwiec 2016.
- [14] L. E. M. M. A. S. J. M. A. Arturo de la Escalera, „Road Traffic Sign Detection and Classification,” *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, grudzień 1997.
- [15] Y. Y. J. G. H. L. Z. Huang, „An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine,” *IEEE Transactions on Cybernetics*, tom 47, nr 4, pp. 920 - 933, 14 marca 2016.
- [16] M. P. a. D. H. T. Ojala, „Performance evaluation of texture measures with classification based on Kullback discrimination of distributions,” w *ICPR*, 1994.
- [17] G. R. Z. W. Y. Z. L. J. G. Wang, „A robust coarse-to-fine traffic sign detection method,” w *Proc. Int. Joint Conf. Neural Netw.*, 2013.
- [18] Z. T. P. P. S. B. P. Dollár, „Integral channel features,” w *Proc. Bri. Conf. Mach. Vis.*, 2009.
- [19] S. L.-A. P. G.-J. G.-M. S. Maldonado-Bascon, „Road-Sign Detection and Recognition Based on Support Vector Machines,” 04 czerwiec 2007.
- [20] R. G. W. E. H. P. D. W. I. M. Creusen, „Color exploitation in hog-based traffic sign detection,” w *International Conference on Image Processing*, Hong Kong, 26-29 wrzesień 2010.
- [21] D. L. J. X. H. Liu, „Real-time recognition of road traffic sign in motion image based on genetic algorithm,” listopad 2002.
- [22] M. J. J. P. Viola, „Robust real-time face detection,” Maj 2004.

- [23] Y. Z. V. R. M. P. T. K. C. Bahlmann, „A system for traffic sign detection, tracking, and recognition using color, shape, and motion information,” w *IEEE Proceedings. Intelligent Vehicles Symposium*, Las Vegas, 6-8 czerwiec 2005.
- [24] T. M. T. B. J. F. C. G. R. Kastner, „Attention-based traffic sign recognition with an array of weak classifiers,” czerwiec 2010.
- [25] A. Z. N. Barnes, „Real-time radial symmetry for speed sign detection,” czerwiec 2004.
- [26] N. B. G. Loy, „Fast shape-based road sign detection for a driver assistance system,” w *IROS*, wrzesień 2004.
- [27] A. I. E. Armingol M. Mata, „Traffic sign recognition and analysis for intelligent vehicles,” *Image and Vision Computing*, tom 21, nr 3, pp. 247-258, marzec 2003.
- [28] R. T. R. B. L. J. V. G. M. Mathias, „Traffic sign recognition-how far are we from the solution?,” w *Proc. Int. Joint Conf. Neural Netw.*, 2013.
- [29] Y. L. P. Sermanet, „Traffic sign recognition with multi-scale convolutional networks,” w *Proc. Int. Joint Conf. Neural Netw.*, 2011.
- [30] A. R. A. M. S. Prieto, „Using self-organising maps in the detection and recognition of road signs,” *Image Vis. Comput.*, tom 6, nr 27, pp. 673-683, 2009.
- [31] S. C. C. F. C. Fang, „Road sign detection and tracking,” w *IEEE Trans. Veh. Technol.*, 2003.
- [32] D. L. S. Z. X. H. B. L. S. H. Zhe Zhu, „Traffic-Sign Detection and Classification in the Wild,” w *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 czerwca 2016.
- [33] Y. L. P. Sermanet, „Traffic sign recognition with multi-scale convolutional networks,” w *Proc. Int. Joint Conf. Neural Netw.*, 2011.
- [34] D. E. X. Z. M. M. R. F. Y. L. P. Sermanet, „Overfeat: Integrated recognition localization and detection using convolutional networks,” w *CoRR*, 2013.
- [35] J. D. T. D. J. M. R. Girshick, „Rich feature hierarchies for accurate object detection and semantic segmentation,” w *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [36] K. v. d. S. T. G. A. S. J. Uijlings, „Selective search for object recognition”, *International Journal of Computer Vision*, w *International Journal of Computer Vision*, 2013.
- [37] P. D. C. L. Zitnick, „Edge boxes: Locating object proposals from edges,” w *ECCV. European Conference on Computer Vision*, wrzesień 2014.
- [38] X. Z. S. R. J. S. K. He, „Spatial pyramid pooling in deep convolutional networks for visual recognition,” *Trans. Pattern Anal. Mach. Intell.*, tom 9, nr 37, pp. 1904-1916, 2015.

- [39] R. B. Girshick, „Fast R-CNN,” w *CoRR*, 2015.
- [40] K. H. R. B. G. J. S. S. Ren, „Faster R-CNN: towards real-time object detection with region proposal networks,” w *CCoR*, 2015.
- [41] C. S. A. T. D. A. D. Erhan, „Scalable object detection using deep neural networks,” w *CoRR*, 2013.
- [42] S. R. D. E. D. A. C. Szegedy, „Scalable high-quality object detection,” w *CoRR*, 2014.
- [43] T. A. Y. Aoyagi, „A study on traffic sign recognition in scene image using genetic algorithms and neural networks,” w *International Conference on Industrial Electronics, Control, and Instrumentation*, Taipei, 9 sierpień 1996.
- [44] B. S. O. H. F. Zaklouta, „Traffic sign classification using K-d trees and random forests,” w *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2011.
- [45] P. Guzik, „Metody wyszukiwania punktów charakterystycznych i wyznaczania ich cech,” *Praca dyplomowa inżynierska Politechniki Warszawskiej*, 2014.
- [46] J. A. M. M. A. de la Escalera, „Traffic sign recognition and analysis for intelligent vehicles,” Division of Systems Engineering and Automation, Madryt, wrzesień 2003.
- [47] I. T. P. A. M. Intell., „Learning to detect objects in images via a sparse part-based representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, tom 26, nr 11, pp. 1475-1490, listopad 2004.