

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

...

Spis treści

Streszczenie	2
Analiza problemu, cel i zakres pracy	2
Metody detekcji obiektów na obrazach panoramicznych.....	4
Algorytmy łączenia obrazów w panoramę	4
Metody klasyczne	4
Metody bazujące na punktach charakterystycznych (kluczowych)	5
Łączenie obrazów	8
Segmentacja obrazów.....	8
Wykrywanie na podstawie koloru	9
Wykrywanie na podstawie kształtu.....	9
Wykrywanie na podstawie tekstury	11
Podejście hybrydowe	12
Algorytm Violi i Jonesa	12
SVM.....	12
Inne podejścia	12
Wnioski	13
Identyfikacja obiektów.....	13
Sieci neuronowe	13
CNN.....	13
SVM.....	14
AdaBoost.....	15
Publiczne bazy danych znaków	15
Program	15
Propozycja rozwinięcia/konstrukcji metody	15
Przygotowanie środowiska badawczego i plan badań	15
Podsumowanie	15
Bibliografia.....	15

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 „Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework” [1]. Dodatkową zaletą pewnych systemów jest możliwość obiektów np. w celu poinformowania kierowców o kursie kolizyjnym oraz zapobiegnięcie wykrycia kilkukrotnie tego samego obiektu. Według autorów pracy „Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey” [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.

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 [3]. Innym podejściem wyznaczenia zbieżności funkcji jest metoda Gaussa-Newtona, Levenberga-Marquardta [4].

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 wzorec 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 [5]. 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 [6] 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 [7]. 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 ogólnosci 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) [8]. 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 wpierw 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 wyłonione 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:

$$\sigma_{x,y} = \frac{\frac{1}{P} \sum_{i=1}^P (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y} \text{ gdzie } P \text{ oznacza liczbę klasyfikowanych obiektów, } \mu \text{ wartości}$$

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 w pierw zostaje wykonane progowanie w określonym kolorze, a następnie rozpoznaje się kształty wyłonię 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

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 [9] 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 [10] 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 [11]. Algorytm z wielu słabych klasyfikatorów tworzy jeden mocny klasyfikator. W przypadku wcześniej wspomnianej pracy [10] 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.

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 [12].

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.

LBP i HOG

LPB (ang. local binary patterns) jest podejściem, które zostało zaprezentowane w 1990, a opisano 1994 roku [13]. Algorytm podczas działania dzieli obraz na mniejsze części np. 16×16 px). W każdej części każdy piksel porównuje się z pikselami sąsiadującymi. W najprostszej wersji jest to 8 pikseli. Następnie w ustalonej kolejności porównuje się każdy piksel sąsiadujący z pikselem centralnym i w zależności czy jego wartość jest większa lub równa czy mniejsza wpisuje się odpowiednio 1 lub 0. Tak powstałe słowo (np. 00101001) zamienia się na liczbę w systemie dziesiętnym. Dzięki takiemu rozwiązaniu gdy zostaje podniesiona jasność całego obrazu, skuteczność algorytmu jest taka sama. W każdej łacie tworzy się histogram. Metoda LPB jest zazwyczaj ściśle połączona z metodą HOG (ang. Histogram Oriented Gradient) dlatego z tak otrzymanych liczb dziesiętnych tworzy się histogram i normalizuje go, a następnie ustala gradient 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 wprowadza się obszar ROI, który zawęża strefę poszukiwań co poprawia szybkość algorytmu. Takie podejście zaproponowano w [14]. Wpierw algorytm ustala potencjalne obszary poszukiwań za pomocą okna przesuwanego o niewielkich rozmiarach. Etap nazywa się filtrowaniem zgrubnym i używa algorytmu HOG z klasyfikacją 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 [15]

Podejście hybrydowe

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

Algorytm Violi i Jonesa

Niekiedy do wykrywania znaków w czasie rzeczywistym stosuje się algorytm Viola–Jones zaproponowany w 2001 przez Paul Viola i Michael Jones do wykrywania twarzy [16]. Wielką zaletą metody jest wykrywanie obiektów w czasie rzeczywistym. 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 obliczenia wprowadza się macierz o rozmiarach rozdzielczości obrazu oryginalnego i oblicza się dla każdej wartości macierzy sumę złożoną z odpowiadającego wartości macierzy piksela i lewej oraz górnej wartości macierzy co pozwala zmniejszyć ilość obliczeń dla każdej cechy Haara. Metoda wykrywania znaków drogowych oraz samochodów a nawet rowerzystów została zaimplementowana w pracy [1]. Metoda zaproponowana w tej pracy oprócz wykrywania znaków wykrywa dwa inne typy obiektów dlatego wprowadzono dodatkowo funkcje subkategoryzacji.

SVM

Podjęcie umożliwiające wyznaczenie obszarów za pomocą maszyny wektorów nośnych przedstawione zostało w pracy [17]. Wpierw system wykrywa obszary występowania znaków za pomocą progowania w przestrzeni HSI, oraz dla znaków białych za pomocą rozkładu achromatycznego. Następnie przy użyciu kilku wytrenowanych maszyn SVM następuje klasyfikacja obszaru. 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. Każdy obiekt znaleziony przez metodę opartą na kolorach wpisywany jest w kwadrat. Następnie wyznacza się 20 odległości obiektu od środka ramki do obiektu i tworzy z nich wektor DtB. Taki wektor dostarczany jest na wejście SVM. Wyznaczony w ten sposób wektor znaków ośmiokątnych niezbyt różni się od znaków okrągłych dlatego klasyfikowany jest dopiero na etapie rozpoznawania znaku. Dużą zaletą metody jest jej odporność na rotację.

Inne podejścia

Wartym wspomnienia podejściem detekcji znaków jest tworzenie mapy cieplnej obszarów występowania znaków. Taka mapa działa na zasadzie algorytmów biologicznych. Autorzy artykułu [18]. [here] (opisać artykuł)

Wybarwianie poprzez szukanie kolorów chromatycznych i achromatycznych jest stosowane w [19] za pomocą prostego filtra wektorowego (SVF). SVF ma cechy, które mogą wyodrębnić określony kolor i wyeliminować wszystkie kontury w tym samym czasie.

Niemniej jednak, niektóre ostatnie prace (patrz [20] i [21]) wdrożyły szybki algorytm oparty na symetrii promieniowej, która jest dostosowana do kształtów trójkątnych, kwadratowych, diamentowych, ośmiokątnych i okrągłych. Działa na gradiencie obrazu w skali szarości i wykorzystuje naturę kształtów, które głośują w punkcie środkowym dla okrągłych znaków i linii głośów 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.

Sieci neuronowe radialne

Wnioski

Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework - Podejścia oparte na fakturze najpierw wyodrębniają ręcznie spreparowane funkcje obliczone na podstawie tekstury obrazów, a następnie wykorzystują te wyodrębnione funkcje do szkolenia klasyfikatora. Popularne ręczne funkcje obejmują HOG, LBP, ACF, itp. [2] , [7] , [11] . Niektóre podejścia [35] , [51] , [63] wykorzystują funkcje HOG z SVM, inne [42] używają funkcji ACF z klasyfikatorem AdaBoost. Oprócz powyższych podejść, spłotowa sieć neuronowa (CNN) jest stosowana do wykrywania znaków drogowych i osiąga doskonałe wyniki w [56] .

Zdecydowanie najbardziej popularne są krawędzie i gradienty, ale zbadano inne opcje, takie jak fale fal HOG i Haar. Etap wykrywania jest zdominowany przez transformatę Hough i jej pochodne, ale dla funkcji falowych HOG i Haar zastosowano również SVM, sieci neuronowe i kaskadowe klasyfikatory.

Identyfikacja obiektów

Sieci neuronowe

Sieci neuronowe były stosowane do wykrywania znaków drogowych od początku. Już artykuł z 1996 roku [12] używa sieci do wykrycia typu znaku drogowego. 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.

CNN

Splotowa sieć neuronowa

sieci splotowe są z natury wydajne, gdy są używane w przesuwanym oknie

[22] P. Sermanet, Y. LeCun, "Rozpoznawanie znaków drogowych za pomocą wieloskalowych sieci splotowych", *Proc. Int. Joint Conf. Neural Netw.* , str. 2809-2813, 2011.

W *OverFeat* [23] Sermanet i in. zauważyli, że sieci splotowe są z natury wydajne, gdy są używane w przesuwanym oknie, ponieważ wiele obliczeń może być ponownie wykorzystanych w nakładających się regionach. Przedstawili sieć, która może określić obwódnię obiektu wraz z etykietą klasy.

Inną szeroko stosowaną strategią wykrywania obiektów za pomocą CNN jest najpierw obliczyć ogólne propozycje obiektów i dokonać klasyfikacji tylko dla tych kandydatów. R-CNN [24] był pierwszym, który użył tej strategii, ale jest bardzo powolny z dwóch powodów. Po pierwsze generowanie propozycji obiektów niezależnych od kategorii jest kosztowne. *Selektywne wyszukiwanie* [25] trwa około 3 s, aby wygenerować 1000 propozycji dla obrazów Pascala VOC 2007; bardziej efektywne podejście *EdgeBoxes* [26] nadal trwa około 0,3 s. Po drugie, stosuje ona głęboką sieć splotową do każdej propozycji kandydackiej, co jest bardzo nieefektywne. Aby poprawić efektywność, sieć gromadzenia piramidy

przestrzennej (SPP-Net) [27] oblicza splotową mapę elementów dla całego obrazu i wyodrębnia wektory elementów ze wspólnej mapy cech dla każdej propozycji. Przyspiesza to podejście R-CNN około 100 razy.

Girshick i in. później zaproponował Fast R-CNN [28], który wykorzystuje warstwę softmax powyżej sieci zamiast klasyfikatora SVM używanego w R-CNN. Zignorowanie czasu oczekiwania na obiekt zajmuje 0,3 s, aby Fast R-CNN mógł przetworzyć każdy obraz. W celu przezwyciężenia wąskiego gardła na etapie propozycji obiektu, w Szybszym R-CNN [29], Ren i in. proponowane *sieci propozycji regionów* (RPN), które wykorzystują splotowe mapy funkcji do generowania propozycji obiektów. Pozwala to generatorowi obiektów na dzielenie się splotami pełnoekranowymi z siecią detekcji, co pozwala systemowi wykrywającemu osiągnąć szybkość 5 klatek na sekundę na wydajnym GPU.

Podczas gdy prace te określają propozycje obiektów ręcznie, Szegedy i in. [30] ulepszyło metodę generowania propozycji opartą na danych [31], a także ulepszyło architekturę sieci, aby osiągnąć liczbę klatek na sekundę 50 fps podczas testów, z konkurencyjną skutecznością wykrywania.

Jednak wydajność wszystkich tych sieci wykrywania obiektów została oszacowana na podstawie VOC i ILSVRC PASCAL, gdzie obiekty docelowe zajmują dużą część obrazu

Głęboka sieć neuronowa

SVM

Część systemów TSR do klasyfikacji znaków drogowych korzysta z maszyny wektorów nośnych w literaturze nazywanej po prostu SVM (z ang. support vector machine). Przykładem takiego systemu jest wcześniej omawiany system z 2013 roku przedstawiony w pracy [14]. Teoria działania systemu SVM polega na znalezieniu hiperpłaszczyzny oddzielającej dwie klasy. Do wytrenowania SVM należy w pierwszym etapie wprowadzić oznaczone dane uczące. Następnie należy znaleźć taką funkcję, która oddzieli inaczej oznaczone klasy. Jeżeli taka funkcja nie istnieje należy wprowadzić funkcję jądra, która przutuje dane na funkcję o wyższym wymiarze. W przypadku wspomnianej 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 [17] 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.

AdaBoost

Klasyfikator AdaBoost używany mi w pracy [1] jako słabe klasyfikatory przyjmuje drzewa decyzyjne

[32]

Y. Aoyagi and T. Asakura, "A study on traffic sign recognition in scene image using genetic algorithms and neural networks, " in Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation, vol. 3, 1996, pp. 1838-1843.

Show Context [View Article Full Text: PDF](#) (560KB) [Google Scholar](#)

10. A. d. l. Escalera, J. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles, " Image and Vision Computing, vol. 21, no. 3, pp. 247-258, 2003.

Show Context [CrossRef](#) [Google Scholar](#)

W [33] opracowano dwie sieci neuronowe (NN), aby wyodrębnić cechy koloru i kształtu. Obie te cechy są stosowane do komponentu odcień i obrazu gradientowego, w którym podane są centra określonych obszarów kolorów i centra pewnych ustalonych kształtów.

Publiczne bazy danych znaków

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

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] 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.
- [4] F. J. Zitová B., „Image registration methods, a survey. Image and Vision Computing,” p. 977–1000, 2003.
- [5] T. Szymczyk, „Metoda dopasowania wzorców w rozpoznawaniu obrazów – ograniczenia, problemy i modyfikacje,” w *Automatyka*, 2008, pp. Tom12, Zeszyt 2.
- [6] 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.
- [7] S. M. Heather J.P., „New adaptive algorithms for real-time registration and fusion of multimodal imagery,” w *Proc. SPIE*, 2010.
- [8] D. Lowe, „Scale-invariant feature transform,” *University of British Columbia*, 1999.
- [9] 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.
- [10] 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.
- [11] Y. F. R. E. Schapire, „Experiments with a New Boosting Algorithm,” w *Machine Learning: Proceedings of the Thirteenth International Conference*, New Jersey, 199.
- [12] 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.
- [13] M. P. a. D. H. T. Ojala, „Performance evaluation of texture measures with classification based on Kullback discrimination of distributions,” w *ICPR*, 1994.
- [14] 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.
- [15] Z. T. P. P. S. B. P. Dollár, „Integral channel features,” w *Proc. Bri. Conf. Mach. Vis.*, 2009.
- [16] M. J. J. P. Viola, „Robust real-time face detection,” Maj 2004.
- [17] S. L.-A. P. G.-J. G.-M. S. Maldonado-Bascon, „Road-Sign Detection and Recognition Based on Support Vector Machines,” 04 czerwiec 2007.

- [18] T. M. T. B. J. F. C. G. R. Kastner, „Attention-based traffic sign recognition with an array of weak classifiers,” czerwiec 2010.
- [19] D. L. J. X. H. Liu, „Real-time recognition of road traffic sign in motion image based on genetic algorithm,” listopad 2002.
- [20] A. Z. N. Barnes, „Real-time radial symmetry for speed sign detection,” czerwiec 2004.
- [21] N. B. G. Loy, „Fast shape-based road sign detection for a driver assistance system,” w *IROS*, wrzesień 2004.
- [22] Y. L. P. Sermanet, „Traffic sign recognition with multi-scale convolutional networks,” w *Proc. Int. Joint Conf. Neural Netw.*, 2011.
- [23] D. E. X. Z. M. M. R. F. Y. L. P. Sermanet, „Overfeat: Integrated recognition localization and detection using convolutional networks,” w *CoRR*, 2013.
- [24] 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.
- [25] 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.
- [26] P. D. C. L. Zitnick, „Edge boxes: Locating object proposals from edges,” w *ECCV. European Conference on Computer Vision*, wrzesień 2014.
- [27] 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.
- [28] R. B. Girshick, „Fast R-CNN,” w *CoRR*, 2015.
- [29] K. H. R. B. G. J. S. S. Ren, „Faster R-CNN: towards real-time object detection with region proposal networks,” w *CCoR*, 2015.
- [30] S. R. D. E. D. A. C. Szegedy, „Scalable high-quality object detection,” w *CoRR*, 2014.
- [31] C. S. A. T. D. A. D. Erhan, „Scalable object detection using deep neural networks,” w *CoRR*, 2013.
- [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] S. C. C. F. C. Fang, „Road sign detection and tracking,” w *IEEE Trans. Veh. Technol.*, 2003.
- [34] P. Guzik, „Metody wyszukiwania punktów charakterystycznych i wyznaczania ich cech,” *Praca dyplomowa inżynierska Politechniki Warszawskiej*, 2014.

