




Rozpoznawanie i zliczanie pojazdów przy użyciu OpenCV





Klasyfikator Haar'a

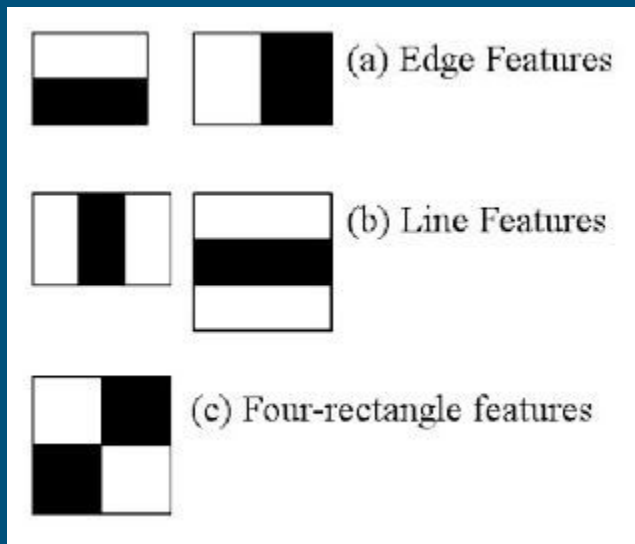
Opis zagadnienia

Podstawy

Wykrywanie obiektów za pomocą klasyfikatorów kaskadowych opartych na cechach Haara jest skuteczną metodą wykrywania obiektów zaproponowaną przez Paula Violę i Michaela Jonesa w ich artykule „Rapid Object Detection using a Boosted Cascade of Simple Features” z 2001 roku. Funkcja kaskady jest szkolona z wielu pozytywnych i negatywnych obrazów. Jest następnie używana do wykrywania obiektów na innych obrazach.

Przypadek wykrywania twarzy

Początkowo algorytm wymaga wielu pozytywnych obrazów (obrazów twarzy) i negatywnych obrazów (obrazów bez twarzy) do szkolenia klasyfikatora. Następnie musimy wyodrębnić z nich cechy. W tym celu używane są cechy Haara pokazane na poniższym obrazie. Każda cecha jest pojedynczą wartością uzyskaną przez odjęcie sumy pikseli pod białym prostokątem od sumy pikseli pod czarnym prostokątem.

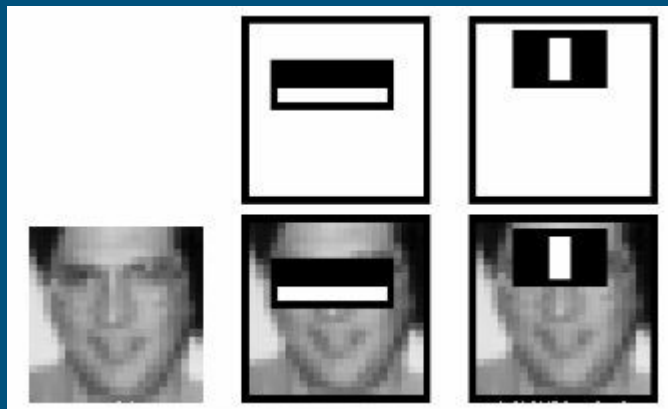


Redukcja obliczeń

Dla każdego obliczenia cechy musimy znaleźć sumę pikseli pod białymi i czarnymi prostokątami. Aby rozwiązać ten problem został wprowadzony tzw. integralny obraz. Jakkolwiek duży jest obraz obliczenia zostają zredukowane dla danego piksela do operacji obejmującej tylko cztery piksele.

Cechy

Większość cech spośród wszystkich, które zostaną wyliczone nie ma znaczenia. Na przykład rozważmy poniższy obrazek. Górny wiersz pokazuje dwie dobre cechy. Pierwsza wybrana cecha w się koncentrować na właściwości, że obszar oczu często jest ciemniejszy, niż obszar nosa i policzków. Druga cecha opiera się na tym, że oczy są ciemniejsze, niż grzbiet nosa. Ale te same okna stosowane do policzków lub innych miejsc są nieistotne.



Adaboost

W tym celu wykorzystujemy każdą cechę na wszystkich obrazach treningowych. Dla każdej cechy zostaje znaleziony najlepszy próg, który klasyfikuje twarze do pozytywnych i negatywnych. Oczywiście pojawią się błędy lub niepoprawne klasyfikacje. Wybieramy cechy z minimalną stopą błędów, co oznacza, że są to cechy, które najdokładniej klasyfikują obrazy twarzy i nie-twarzy. Każdy obraz ma na początku taką samą wagę. Po każdej klasyfikacji zwiększa się waga błędnie sklasyfikowanych obrazów. Następnie wykonywany jest ten sam proces. Obliczane są nowe poziomy błędów oraz nowe wagi. Proces jest kontynuowany do momentu osiągnięcia wymaganej dokładności lub stopy błędów albo do znalezienia wymaganej liczby cech.

Wynik trenowania

Ostatecznie uzyskany klasyfikator to ważona suma słabych klasyfikatorów. Nazywa się je słabym, ponieważ same nie mogą sklasyfikować obrazu, ale razem z innymi tworzą silny klasyfikator. Nawet 200 cech zapewnia wykrywanie z 95% dokładnością.

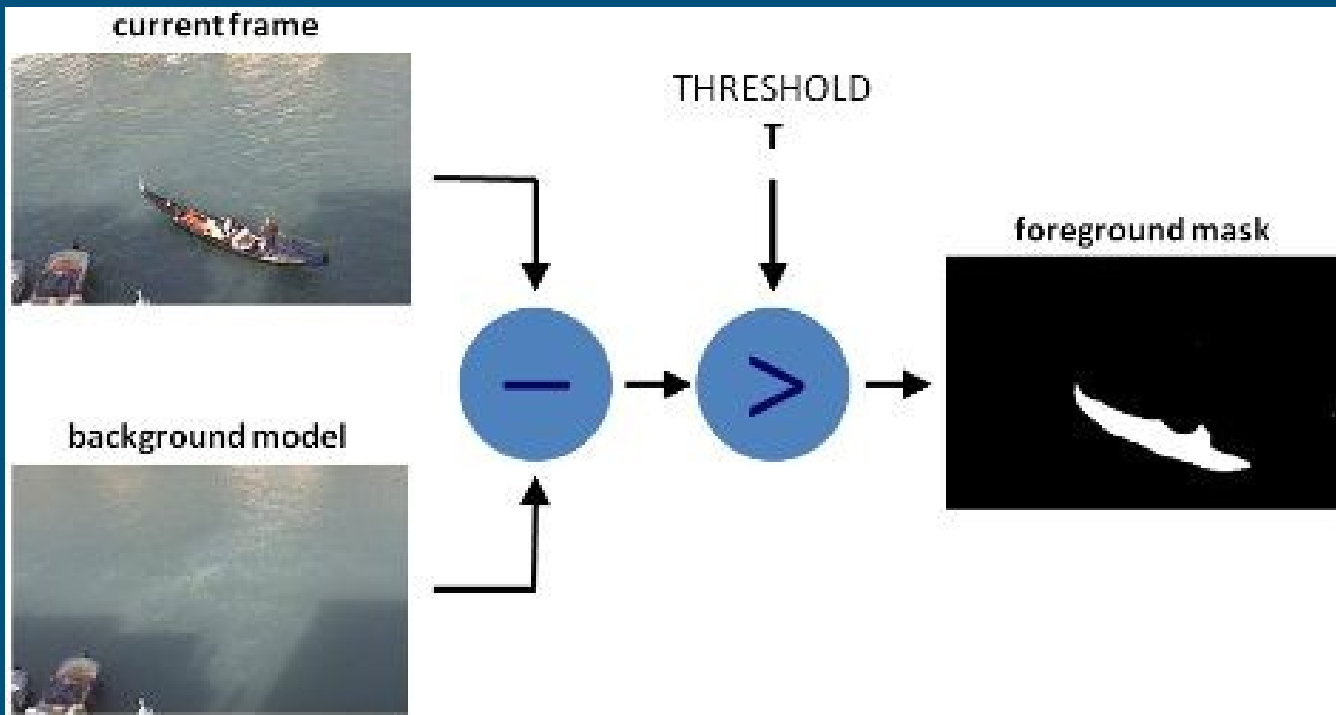
Optymalizacja klasyfikacji

W obrazie większość obszaru nie jest twarzą. Dzięki temu sprawdzamy po prostu, czy okno jest obszarem twarzy, jeśli nie to odrzucamy automatycznie i nie przetwarzamy ponownie. Staramy się skupiać na regionach, w których może być twarz. W ten sposób spędzamy więcej czasu na sprawdzaniu możliwych obszarów twarzy.

W tym celu została wprowadzona koncepcja kaskady klasyfikatorów (*Cascade of Classifiers*). Zamiast stosować wszystkie 6000 cech w oknie, cechy są pogrupowane w różne etapy klasyfikatora i stosowane pojedynczo (zwykle pierwsze kilka etapów będzie zawierało znacznie mniej cech). Jeśli sprawdzanie nie powiedzie się na pierwszym etapie to zostaje odrzucone. Nie bierzemy pod uwagę pozostałych cech. Jeśli przejdzie to zostanie zastosowany drugi etap cech i kontynuacja procesu. Okna, które przechodzi wszystkie etapy będzie obszarem twarzy.

Background Subtraction (BS)

- główny krok wstępnego przetwarzania obrazu
- wyodrębnia ruchomy pierwszy plan ze statycznego tła
- zastosowanie do wyodrębniania wchodzących osób do pomieszczenia, poruszających się pojazdów na drodze
- radzi sobie z poruszającymi się cieniami pojazdów



$$\text{foreground_objects} = \text{current_frame} - \text{background_layer}$$

źródło:

<https://medium.com/machine-learning-world/tutorial-making-road-traffic-counting-app-based-on-computer-vision-and-opencv-166937911660>

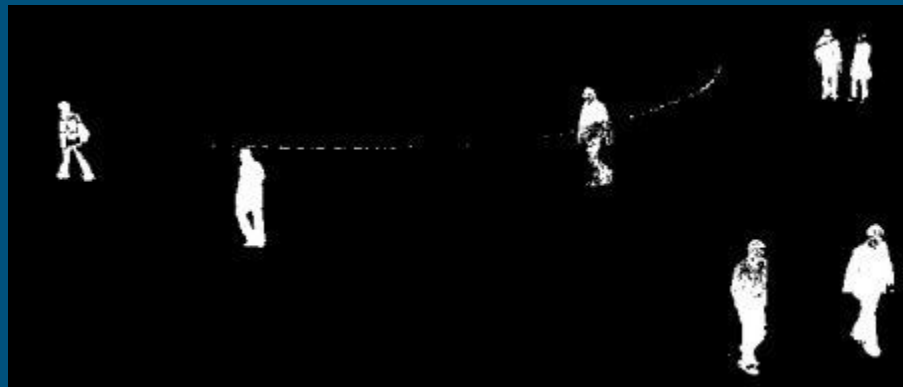
Algorytmy Background Subtraction w OpenCV

- BackgroundSubtractorMOG
- BackgroundSubtractorMOG2
- BackgroundSubtractorGMG

Background Subtraction - przykład



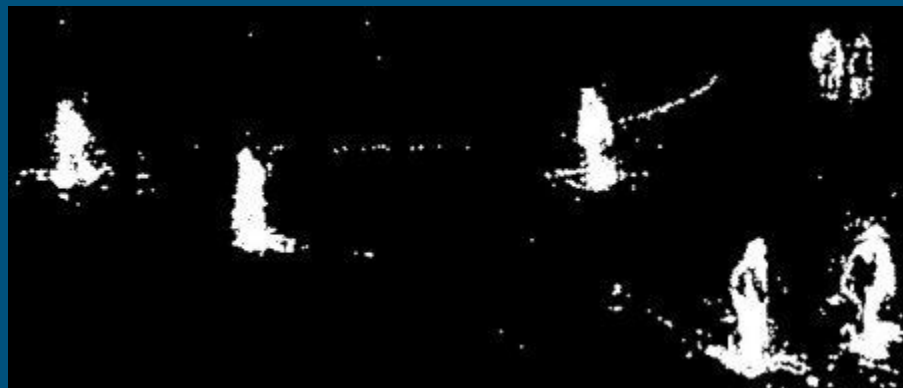
oryginalny obraz



BackgroundSubtractorMOG



BackgroundSubtractorMOG2



BackgroundSubtractorGMG

źródło:

https://docs.opencv.org/4.0.1/db/d5c/tutorial_py_bg_subtraction.html



z lewej strony oryginalne zdjęcie, z prawej po zastosowaniu BackgroundSubtractorMO

źródło:

<https://medium.com/machine-learning-world/tutorial-making-road-traffic-counting-app-based-on-computer-vision-and-opencv-166937911660>

OpenCV image filters

- usunięcie szumu z maski pierwszego planu
- operacja zamknięcia do usuwania luk w obszarach
- operacja otwarcia do usuwania punktów 1-pikselowych
- dylacja do pogrubiania obiektów



źródło:

<https://medium.com/machine-learning-world/tutorial-making-road-traffic-counting-app-based-on-computer-vision-and-opencv-166937911660>

Object detection by contours

OpenCV: `cv2.findContours`