

Maksymilian Zgierski, Krzysztof Lichy

Instytut Informatyki Politechnika Łódzka

Wólczańska 215 , 90-924 Łódź

email: krzysztof.lichy@p.lodz.pl

WYKORZYSTANIE OBRAZU Z KAMERY DO ANALIZY NATEŻENIA RUCHU DROGOWEGO

Streszczenie –W artykule opisano autorską aplikację wykorzystującą obraz z kamery do analizy natężenia ruchu drogowego. Aplikacja na podstawie obrazu jest w stanie oszacować średnią prędkość poruszających się obiektów. Aplikacja działa w czasie rzeczywistym ma niewygórowane wymagania sprzętowe. Przedstawiono wnioski z jej działania oraz perspektywy wdrożenia i rozwoju.

Słowa kluczowe: natężenie ruchu drogowego, analiza obrazu

1 Wstęp

W życiu codziennym, każdy człowiek jest zmuszony do podróżowania przez miasto w celach służbowych. Niejednokrotnie zdarza się, że z powodu nadmiernego natężenia ruchu dochodzi do spóźnień. W większości takich sytuacji, istnieje kilka możliwych dróg, aby dotrzeć do punktu docelowego, a najbardziej popularne ulice są zazwyczaj najmniej przejezdne. W przypadku przeciętnego obywatela spędzanie czasu w korkach drogowych zwykle nie ma znaczących negatywnych skutków, jednak w przypadku służb specjalnych jak pogotowie czy straż pożarna niedopuszczalnym jest marnowanie czasu. Biorąc to pod uwagę, celem będzie stworzenie aplikacji, która przeanalizuje obraz z kamery (np. kamery monitoringu miejskiego) i na jego podstawie oszacuje średnią prędkość poruszających się pojazdów, a następnie porówna otrzymaną wartość z maksymalną dopuszczalną prędkością na danej ulicy. W ten sposób można ocenić aktualne zatłoczenie na ulicy. Aplikacja może stanowić rzetelne źródło danych dla wszystkich użytkowników dróg, którzy są zainteresowani aktualnym stanem zatłoczenia miasta. Niezbędnym jest, aby aplikacja sprawnie rozpoznawała wielkość poruszających się obiektów, aby odróżnić poruszające się samochody od pieszego przechodzącego przez ulicę, biegnącego zwierzęcia lub latających liści na wietrze. Te zabiegi są potrzebne, aby wyeliminować fałszywe wyniki, które mógłby produkować program. Kolejnym ważnym aspektem działania aplikacji jest szybkość przetwarzania danych. Istotne jest to, żeby działała ona w czasie rzeczywistym i jej wyniki mogły być

na bieżąco pokazywane na tablicach informacyjnych w mieście, na stronie internetowej lub aplikacji mobilnej. Kluczowym elementem będzie znalezienie odpowiedniej równowagi pomiędzy rozdzielczością analizowanych obrazów oraz wielkością analizowanego obszaru drogi, a możliwościami sprzętowymi. Zagadnienie zakorkowania ulic w mieście było już wielokrotnie podejmowane i kilka firm podjęło się rozwiązania tego problemu. Najbardziej znanym rozwiązaniem jest Google Maps, które oferuje użytkownikom podgląd zatłoczenia ulic. Popularnością cieszą się również rozwiązania TomTom Traffic lub nawigacja Targeo (występuje głównie w Polsce). Jednak wszystkie z wyżej wymienionych programów mają swoje wady, których pozbawiona będzie opisywana aplikacja.

Aplikacja została napisana w języku Java. Do przechowywania danych zastosowano bazę danych PostgreSQL. Dodatkowo wykorzystano bibliotekę OpenCV, która służy do przetwarzania obrazów. Jest w niej zaimplementowanych wiele metod do obróbki oraz manipulacji obrazów. Posiada ona między innymi narzędzia wspomagające realizację wcześniej wspomnianych algorytmów do wykrywania poruszających się obrazów. Dalsza analiza prędkości obiektów oraz ich rozmiar są określane na podstawie własnych metod, bez wykorzystania wcześniej zaimplementowanych rozwiązań. Na pierwszym miejscu w hierarchii ważności stawiane będą wyniki jak najbliższe rzeczywistości, wydajność pozostawiając na drugim planie. W związku z tym, do celów testowych zostaną wykorzystane filmy nagrane w małej rozdzielczości oraz dodatkowo będzie ograniczony analizowany obszar klatki filmowej (nie trzeba analizować całej klatki filmu).

2 Zalety tworzonej aplikacji

Podstawową korzyścią, która będzie płynąć z tworzonej aplikacji jest oczywiście analiza natężenia ruchu drogowego. Podstawową różnicą w porównaniu do istniejących rozwiązań jak na przykład Google Maps jest brak konieczności posiadania przez uczestników ruchu urządzeń mobilnych z włączonym odbiornikiem GPS oraz połączeniem z Internetem.

Kolejną korzyścią są dane statystyczne, które będą zapisywane w bazie danych. Aplikacja automatycznie będzie odnotowywać wyprodukowane rezultaty, które w przyszłości mogą mieć znaczenie dla działań urzędów miejskich, na terenie których zostanie wdrożona aplikacja. Informacje pobrane z bazy danych tworzonego programu mogą się przydać przy planowaniu zmian organizacji ruchu w miastach, planowaniu imprez masowych lub przy remontach drogowych.

3 Wady istniejących rozwiązań

Główną wadą Targeo jest fakt, że obejmuje ona swoim zasięgiem głównie drogi znajdujące się w Polsce. W porównaniu do istniejących już rozwiązań jest to znaczne ograniczenie. Twórcy starają się wprowadzać mapy innych państw, jednak na obecnym etapie rozwoju aplikacji służących do nawigacji i informowania o stanie dróg, zaistnienie na zagranicznych rynkach będzie bardzo dużym wyzwaniem.

Również wygląd interfejsu użytkownika, który jest nieprzyjemny dla oka w porównaniu z Google Maps lub TomTom Traffic. Istnieje cała gama gotowych szablonów (dotyczy głównie stron internetowych) lub widgetów (dotyczy głównie aplikacji mobilnych), które umożliwiłyby znaczną poprawę estetyki.

Targeo przez pierwsze kilka lat swojej działalności rozwijało głównie witrynę internetową, zapominając całkowicie o aplikacjach dedykowanych na urządzenia mobilne. Obecnie, kiedy użytkownicy mają już swoje przyzwyczajenia dotyczące map na systemach Android lub iOS, prawie niemożliwym staje się wyróżnienie wśród konkurencji. [8]

TomTom Traffic oferuje bardzo przyjemny interfejs graficzny oraz stosunkowo rzetelne informacje dotyczące natężenia ruchu. Niestety, aby korzystać z tej aplikacji niezbędne jest posiadanie osobnego urządzenia, produkowanego przez tę firmę lub korzystać z płatnych aplikacji mobilnych na systemy Android, iOS oraz Windows Phone. Użytkownicy osobnej nawigacji formy TomTom zmuszeni są oprócz opłaty za urządzenie, uiszczać opłatę za aktualizację map. Jest to niezbędne, aby mieć aktualne informacje o stanie dróg, zwłaszcza, gdy żyjemy w kraju, który szybko rozwija infrastrukturę drogową. Osoby, które zdecydowały się na korzystanie aplikacji Mobile GO (aplikacja firmy TomTom, która zawiera usługę TomTom Traffic) są zobowiązani do płacenia odgórnie narzuconego abonamentu (miesięcznego lub rocznego) lub opłaty są generowane za każdy przebyty kilometr (lub milę – zależy od kraju). Wszystkie opisane funkcjonalności są oferowane przez Google Maps całkowicie za darmo. Pomimo tych niedogodności opisywana aplikacja znajduje zwolenników na całym świecie, jednak można założyć, że opłaty z pewnością zmniejszą grono potencjalnych odbiorców. [6]

Jedną z poważniejszych wad jest to, że wyżej opisywane aplikacje wymagają, aby użytkownicy mieli włączone lokalizowanie w urządzeniu mobilnym oraz dostęp do Internetu. Najczęściej, zwłaszcza poruszając się po ulicy, dostęp do sieci Wi-Fi jest niemożliwy, więc aby sprostać wymogom konieczności dostępu do Internetu korzystamy z własnych zasobów, za które musimy płacić (pomijamy szczególne rodzaje umów z

operatorami telefonii komórkowych, gdzie mamy nielimitowany dostęp do Internetu). Dostęp do danych dotyczących lokalizacji może ingerować w prywatność wielu osób. Niestety, aby informacje o natężeniu ruchu drogowego były rzetelne, duża ilość użytkowników musi regularnie przysyłać dane o swojej lokalizacji. W większości przypadków jest to robione nieświadomie, ponieważ twórcy systemów operacyjnych, do realizacji swoich potrzeb, automatycznie uruchamiają cykliczne przysyłanie informacji o położeniu urządzeń mobilnych. Przykre jest to, że wszyscy korzystając z urządzeń elektronicznych zmuszeni są do udostępniania informacji na swój temat (w tym przypadku lokalizacji, ale powszechnie wiadomo, że nie tylko takie dane są pobierane z telefonów czy tabletów), a koncerny informatyczne takie jak Google, Apple czy TomTom nie pytają nas o zgodę. W rezultacie, jedynym sposobem, aby ograniczyć wysyłanie informacji o lokalizacji jest wyłączenie odbiornika GPS lub dostępu do Internetu, które są niezbędne w większości współczesnych aplikacji.

Ostatnią, równie poważną wadą, jest problem z określeniem środka transportu, którym użytkownicy drogi się poruszają. Nawet poprzez udostępnianie informacji dotyczących lokalizacji urządzeń mobilnych, nie można uzyskać pewności, że dane dotyczące przejeźdźności dróg będą rzetelne. Odbiorniki GPS w urządzeniach mobilnych są w stanie określać położenie z dokładnością do 2-3m. W efekcie, odróżnienie pieszego, kierowcę samochodu, motocyklu i osobę poruszającą się rowerem jest kłopotliwe. Zakładając, że pomiędzy autami stojącymi na światłach przejeżdża grupa motocyklistów, nie jest możliwe, aby ustalić, czy rzeczywiście w danym miejscu jest problem z przepływem ruchu, ponieważ połowa pojazdów stoi nieruchomo, a połowa bez problemów przejeżdża. Podobny problem jest z cyklistami jadącymi przy skraju drogi lub pieszymi, którzy np. uprawiają jogging. Co prawda, Google i inne konkurencyjne firmy zapewniają, że dysponują zaawansowanymi algorytmami, które pozwalają określić, jakim środkiem transportu porusza się dany użytkownik, ale rzeczywistość pokazuje, że nawet tak popularne rozwiązania jak np. Google Maps miewa problemy z określeniem natężenia ruchu na danym odcinku drogi oraz te informacje pojawiają się często z nawet kilkuminutowym opóźnieniem.

Biorąc powyższe wady pod uwagę, idealnym rozwiązaniem byłaby aplikacja, która nie wymagałaby od większości użytkowników drogi posiadania urządzenia mobilnego, a tym bardziej nie narzucałaby konieczności posiadania działającego odbiornika GPS oraz włączonej transmisji danych po sieci. Dodatkowo pożądanym by było, aby nastąpiło automatyczne rozpoznawanie środka transportu, czyli piesi oraz osoby poruszające się rowerami lub innymi rekreacyjnymi pojazdami nie powinny być brane pod uwagę przy analizie natężenia ruchu drogowego. Motocykle mogą być opcjonalne, a decyzja powinna

należać do operatora aplikacji. Jedynie obiekty minimum czterokołowe muszą być bezwzględnie analizowane.

4 Biblioteka opencv

OpenCV jest skrótem od Open Source Computer Vision. Jest to biblioteka zajmująca się przetwarzaniem obrazu oraz uczeniem maszynowym. Można z niej korzystać na podstawie licencji BSD, co oznacza, że występuje możliwość nie tylko modyfikacji oryginalnego kodu źródłowego oraz jego rozprowadzania po modyfikacjach, ale również rozprowadzanie programu bez kodu źródłowego, pod warunkiem zamieszczenia w programie informacji o autorach biblioteki.

Istnienie OpenCV zostało zapoczątkowane przez firmę Intel, jednak aktualnie za rozwój oprogramowania jest odpowiedzialne osobne przedsiębiorstwo Itseez. Biblioteka została napisana w języku C, ale w późniejszym czasie powstało wiele nakładek do innych języków programowania i obecnie możemy z niej korzystać również w Javie, Pythonie, C++, C# oraz programie MATLAB. Oprócz oczywistego wsparcia dla platform Windows oraz Linuks, pozwolono również wykorzystywać bibliotekę na Androidzie i systemie Mac OS, co znacząco zwiększa spektrum zastosowań.

Bibliotekę składa się z ponad 2500 algorytmów, które zostały napisane w jak najbardziej optymalny sposób, pozwalając na wydajne przetwarzanie obrazu. Wszystkie te metody mogą zostać wykorzystane do takich zadań jak wykrywanie i rozpoznawanie twarzy, identyfikowanie obiektów, rozpoznawanie i klasyfikowanie ludzkich zachowań na nagraniach wideo, śledzenie poruszających się obiektów, tworzenie modeli 3D obiektów, łączenie obrazów w celu otrzymania zdjęć o wysokiej rozdzielczości, znajdowanie obrazów podobnych do zdjęć znajdujących się w bazie danych, usuwanie efektu czerwonych oczu ze zdjęć, śledzenie ruchów oczu, rozpoznawanie miejsc i scenarii oraz wspomaganie umieszczania znaczników i obiektów w programach wykorzystujących techniki rozszerzonej rzeczywistości. Ten ogrom możliwości spowodował, że OpenCV stało się najbardziej popularną biblioteką służącą do przetwarzania obrazu. Na oficjalnej stronie projektu znajduje się informacja, że ilość użytkowników przekroczyła 47 tysięcy, a sama biblioteka została pobrana ponad 7 milionów razy. Warto dodać, że OpenCV jest wykorzystywane nie tylko do projektów naukowych lub uczelnianych. Takie firmy jak Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota postanowiły użyć jej w swoich komercyjnych produktach. Dzięki łatwości w zastosowaniu, biblioteka jest stosowana również w wielu start-upach, np. Applied Minds, VideoSurf, and Zeitera. Przykładowe zastosowania OpenCV to analiza obrazu z kamer miejskich i wykrywanie nieodpowiednich zachowań, np.

zamieszek (zastosowane w Izraelu), monitorowanie sprzętu w kopalniach (zastosowane w Chinach), wspomaganie oprogramowania w robotach służących do podnoszenia i przynoszenia przedmiotów (firma Willow Garage), wspomaganie nadzorowania basenów i informowanie w przypadku, gdy topią się ludzie (zastosowane w wielu krajach Europy), wspomaganie oprogramowania wykorzystywanego w sportach biegowych (zastosowane w Hiszpanii i Nowym Jorku), sprawdzanie czy na autostradzie nie znajdują się niechciane przedmioty (zastosowane w Turcji), sprawdzanie etykiet na produktach w magazynach, rozpoznawanie twarzy w czasie rzeczywistym (zastosowane w Japonii).

Biorąc pod uwagę możliwości biblioteki, wydaje się ona być idealnym narzędziem, które pozwoli zaoszczędzić czas przy tworzeniu aplikacji do analizy natężenia ruchu drogowego. Co więcej, optymalnie zaimplementowane algorytmy sprawią, że program będzie dużo wydajniejszy. W kontekście tej pracy interesujące będą jedynie metody odpowiedzialne za wykrywanie ruchu na filmie. Należy się liczyć z tym, że na klatkach mogą pojawić się biegnący pies, szybko przechodzący człowiek lub liść lecący na wietrze, a są to obiekty, których nie powinno się uwzględniać w analizie. W związku z tym, za pomocą własnych rozwiązań trzeba określić dokładne położenie pojazdów (klasyfikowanie obiektu do kategorii pojazdów odbywa się na podstawie rozmiaru) [3].

5 Wymagania funkcjonalne aplikacji

Zadaniem aplikacji jest analizowanie natężenia ruchu drogowego na podstawie filmu. Jest to realizowane poprzez porównywanie prędkości wybranych poruszających się obiektów z maksymalną dopuszczalną prędkością na danej ulicy. W związku z powyższym program musi spełniać kilka bardzo ważnych wymagań.

Pierwszą z funkcjonalności jest wykrywanie poruszających się obiektów na nagraniu. Aby to osiągnąć należy zastosować jeden z opisanych wcześniej algorytmów służących do oddzielania pierwszego planu od tła. Jednak, czarno-biały obraz wymaga dalszej obróbki. Jak łatwo się domyślić, w słoneczne dni mogą zdarzać się chwilowe refleksy świetlne, które mogą imitować występowanie poruszających się obiektów tam, gdzie one w rzeczywistości nie występują. Podobną sytuację można zaobserwować, gdy ulica będzie mokra od deszczu, a światła reflektorów z łatwością będą się odbijały. Należy zauważyć, że takie niechciane efekty są najczęściej bardzo małych rozmiarów i są zwykle oddalone od ruchomego pojazdu. Refleksy występujące na powierzchni obiektu, który jest w kręgu zainteresowań są neutralne.

Kolejną funkcjonalnością będzie określenie granic poruszającego się obiektu. Oznacza to, że aplikacja powinna być w stanie narysować prostokąt wokół poruszającego się pojazdu, pomijając znajdujące się w

pewnej odległości od niego artefakty jak refleksy świetlne czy liście latające na wietrze.

Trzecią funkcjonalnością będzie rozpoznawanie wielkości ruchomych obiektów, aby na podstawie przybliżonego rozmiaru określić rodzaj pojazdu. Na pewno analizie prędkości będą podlegać samochody osobowe oraz ciężarowe. Jednak rowerzyści i motocykliści mogą, ale nie muszą być brani pod uwagę przy obliczaniu natężenia ruchu drogowego. Decyzję o tym należy pozostawić osobie, która będzie obsługiwać aplikację.

Następna funkcjonalność będzie związana z wykrywaniem przez aplikację, kiedy film jest nieodpowiedni do analizy natężenia ruchu. Taka sytuacja może występować na przykład w bardzo wietrzne dni, kiedy kamera zostanie poruszona. Niezbędne jest, aby w czasie rejestrowania materiału wideo, sprzęt nagrywający był zainstalowany w taki sposób, które wyeliminuje drgania. W przypadku, gdy program wykryje niechciane ruchy kamery powinien wyświetlić stosowny komunikat.

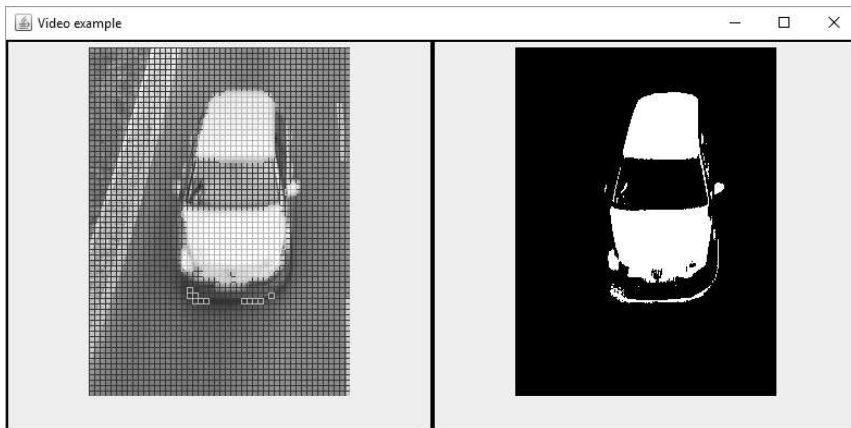
Ostatnią funkcjonalnością będzie zapis danych statystycznych do bazy danych. W trakcie przetwarzania filmów, aplikacja co pewien czas (domyślnie 5 minut) powinna utrzymywać informacje o ilości oraz wielkości spostrzeżonych pojazdów. Te dane mogą być przydatne przy późniejszym planowaniu różnych zdarzeń drogowych, które mogą mieć miejsce w punkcie, w którym znajduje się sprzęt nagrywający, z którego pochodzi materiał filmowy.

6 Proces implementacji

Pierwszym krokiem, który należy zawrzeć w implementacji jest wykrywanie poruszających się obiektów. Ze względu na odpowiednią precyzję i niezbyt duże zapotrzebowanie na moc obliczeniową zostanie użyty algorytm Mixture of Gaussians. Wykorzystanie MOG z biblioteki OpenCV jest rzeczą naturalną, ze względu na optymalizację aplikacji. Aby wykryć pojazdy należące do pierwszego planu wystarczy wywołać kilka linii kodu. Poniżej znajduje się tekst klasy MOG, w której jest metoda bsMOG, która odpowiada za wykorzystanie algorytmu na kolejnej klatce filmu.

```
public class MOG {
    private static Mat fgMaskMOG = new Mat();
    private static BackgroundSubtractorMOG pMOG = new
        BackgroundSubtractorMOG();
    public static BufferedImage bsMOG(Mat current) {
        pMOG.apply(current, fgMaskMOG);
        return Tools.createAwtImage(fgMaskMOG);
    }
    ...
}
```

Kolejnym krokiem jest wykrycie powierzchni obiektu. W tym celu należy podzielić klatkę, na której został użyty algorytm Mixture of Gaussians na małe kwadraty. W przypadku filmów testowych, wymiary obrazu wynoszą 480 na 640 pikseli, co pozwala na podział na figury o wymiarach 5 na 5 pikseli. Następnym krokiem jest wykrycie skupisk kwadratów, wewnątrz których było więcej niż 16 białych pikseli. Wartość 16 można zamienić na dowolną inną liczbę, w tym przypadku została ona dobrana doświadczalnie. Istotne jest to, aby to były całe skupiska, a nie pojedyncze figury. Aby to osiągnąć należy wziąć losowo jeden kwadrat, który spełnia warunek 16 białych pikseli, a następnie sprawdzać czy kolejne kwadraty, które spełniają ten sam warunek znajdują się w pobliżu pierwotnie wybranej figury. Poniżej znajduje się rysunek obrazujący efekt opisywanych działań.



Rys. 1. Przykład działania aplikacji

Następnie, należy obrysować znalezione skupiska prostokątem. W tym celu analizuje się skrajne współrzędne wszystkich zielonych kwadratów i znajduje się koordynaty brzegowe dużego prostokąta, który w aplikacji jest reprezentowany przez klasę Rectangle. Aby określić prędkość, z jaką porusza się pojazd, należy porównać położenie obramowania pojazdu na przestrzeni kolejnych klatek. Łatwo obliczyć ilość pikseli, o które obiekt się przesunął. Aby odnieść to do wielkości rzeczywistych, należy znać przybliżoną wysokość, na jakiej znajduje się kamera oraz długość odcinka drogi, która jest obserwowana. Na podstawie tych informacji, można dokonać łatwych przekształceń matematycznych, aby otrzymać efekt, jak ten pokazany na zdjęciu poniżej.



Rys. 2. Przykład działania aplikacji – obliczanie prędkości obiektu

Następnie warto porównać wielkość otrzymanego prostokąta z wielkością zdefiniowaną w ustawieniach. Jeżeli niebieska figura jest większa lub równa tej z ustawień, to powinna być uwzględniona w analizie i wynik powinien mieć wpływ na natężenie ruchu drogowego na danej ulicy oraz informacje o tym powinny się znaleźć w danych statystycznych w bazie danych. W przeciwnym razie, poruszający się obiekt należy pominąć.

Ostatnim krokiem będzie porównanie otrzymanej prędkości z maksymalną dopuszczalną szybkością na danej ulicy. Na podstawie tego porównania należy określić poziom natężenia ruchu. Jeżeli statystycznie pojazdy poruszają się z prędkością zbliżoną do limitu szybkości, należy uważać, że ruch na ulicy jest niewielki. Jeżeli jednak zwykle prędkość obiektów jest dużo niższa niż maksymalna dozwolona, oznacza to, że występuje problem z przepływem ruchu i przejazd daną ulicą może być utrudniony.

7 Testy aplikacji

Testy eksperckie są zbiorem sprawdzianów, którym powinna zostać poddana aplikacja. Będą one przeprowadzone przez grupę ekspertów posiadających odpowiednie kwalifikacje. Idea polega na nagraniu odpowiedniej ilości materiału wideo w różnorodnych warunkach zarówno pogodowych, jak i natężenia ruchu, a następnie skonfrontowania wyników wyprodukowanych przez program z rezultatami podanymi przez zespół specjalistów. Program powinien zostać zbadany pod kątem nieomyślności wszystkich wymagań funkcjonalnych przedstawionych wcześniej. W przypadku aplikacji eksperci powinni dokładnie przeliczyć ilość oraz rodzaj poruszających się w polu widzenia kamery pojazdów, a następnie skonfrontować swoje rachunki z tym, co będzie widać na

ekranie. Począwszy od poprawnego wykrycia poruszającego się obiektu, poprzez umiejętność odpowiedniego obrysowania pojazdu prostokątem, przypisanie do określonego rodzaju pojazdów na podstawie wielkości prostokąta, na określaniu prędkości kończąca.

8 Testy wydajnościowe

Przez testy wydajnościowe należy rozumieć wydajność aplikacji analizującej natężenie ruchu drogowego, czyli ilość klatek filmowych, które mogą zostać przetworzone w ciągu jednej sekundy. Oczywiście wielkość obrazu ma największy wpływ na wydajność programu. Jak zostało wspomniane w jednym z wcześniejszych rozdziałów, wydajność nie była priorytetem przy tworzeniu aplikacji, a osiągnięcie poprawnego wyniku, czyli obliczenie odpowiedniej prędkości pojazdu, która później w bardzo łatwy sposób może być porównana z maksymalną dopuszczalną szybkością na danej ulicy. Pomimo ustalonych priorytetów, w czasie programowania poświęcono czas, aby algorytmy zostały napisane w sposób względnie optymalny. Nie bez znaczenia okazało się również wykorzystanie biblioteki OpenCV, w której, jak zapewniają twórcy, metody detekcji poruszających się obiektów zostały zaimplementowane w jak najbardziej wydajny sposób. Niemniej, z pewnością istnieje kilka sposobów na zwiększenie możliwości obliczeniowych. Najprostszą techniką optymalizacyjną, którą można zastosować jest określenie obszaru w obrazie, który powinien być analizowany. Pomimo faktu, że kamera znajduje się w punkcie bezpośrednio nad ulicą, pole widzenia może objąć również fragmenty trawnika, krawężniki lub torowisko tramwajowe, czyli elementy infrastruktury, gdzie mamy pewność, że interesujące nas pojazdy nie będą się poruszać. Pozwoli to na nieprzetwarzanie przynajmniej kilku tysięcy pikseli. Na testowych filmach został użyty film o rozmiarze klatki 640 na 480 pikseli, co daje nam w sumie 307 200 pikseli, a więc oszczędność czasu poprzez zmniejszenie obszaru analizy może okazać się bezcenne przy masowym wykorzystaniu aplikacji.

Procesor Intel i5 z taktowaniem zegara na poziomie 2,5GHz pozwala aplikacji na przetwarzanie do maksymalnie 15 klatek na sekundę. Pomiar nie był wykonywany przy pomocy narzędzi zewnętrznych, a jedynie poprzez wykorzystanie metod języka Java pozwalających na pomiar czasu. Jest to wynik zadowalający na etapie prototypu. Warto byłoby również sprawdzić jakość wyników, które produkowałby program, gdyby wielkość klatki filmowej została zmniejszona. Nie jest wykluczone, że wyniki również pokrywałyby się z rzeczywistością, a wydajność z całą pewnością drastycznie by wzrosła.

9 Wnioski

W ramach aplikacji udało się wykonać implementację sprawdzania prędkości poruszających się pojazdów. Sprawdzanie wielkości obiektu, porównywanie obliczonej prędkości z maksymalną dozwoloną wartością oraz zapisywanie danych statystycznych do bazy danych zostaną zaimplementowane w wersji rozwojowej oprogramowania. Biblioteka OpenCV udostępnia metody implementujące algorytm wykrywania pierwszego planu za pomocą algorytmu Mixture of Gaussians.

Zauważyć należy, że można ograniczyć ilość czasu spędzanego w korkach, co w efekcie przekłada się na korzyści materialne (jak zaoszczędzenie pieniędzy na niższym spalaniu paliwa w pojazdach, co w skali całego miasta jest dużą zaletą, poświęcenie czasu na pracę zamiast na stanie w korku drogowym), ekologiczne (zmniejszone spalanie oznacza mniejszą emisję spalin) oraz zdrowotne (brak zatłoczonych ulic oznacza poprawę kondycji psychicznej mieszkańców oraz zmniejszenie poziomu stresu). Dodatkowo usprawniony ruch samochodów spowoduje poprawiony czas działania służb specjalnych jak pogotowie, policja czy straż pożarna oraz ułatwi życie pracownikom wielu zawodów jak na przykład taksówkarz, kurier czy zawodowy kierowca samochodu ciężarowego. Istnieją co prawda rozwiązania, które starają się obliczać natężenie ruchu drogowego, jednak ograniczenia aplikacja jak Google Maps czy TomTom Traffic są na tyle duże, że korzystanie z nich w czasie jazdy samochodem może być nieporęczne. Co więcej, aby wyniki były rzetelne, wiele osób musi przesyłać sukcesywnie informacje o swoim położeniu, co wiąże się z koniecznością posiadania włączonego odbiornika GPS oraz dostępem do Internetu. Przedstawione rozwiązanie jest pozbawione tych wad, ponieważ wyniki produkowane przez aplikację mogą być wyświetlane na tablicach elektronicznych znajdujących się na terenie całego miasta oraz nie wymaga od użytkowników drogi posiadania włączonych systemów, które zostały wymienione powyżej.

Z całą pewnością, aplikacja wymaga zoptymalizowania. Obecnie, program jest w stanie analizować film z prędkością do 15 klatek na sekundę, co nie jest najlepszym wynikiem. Aby wyniki były dostarczane w czasie rzeczywistym, potrzebne jest co najmniej 20 klatek na sekundę. Żeby to uzyskać można zdefiniować obszar, który powinien podlegać przetwarzaniu. Spowoduje to zmniejszenie ilości pikseli, które będą musiały zostać poddane operacji analizy przy niezmięnionej jakości filmu. Można założyć, że pojazdy jadące jednym pasem ruchu posiadają określoną szerokość. Jednak kamera, której zadaniem jest tworzenie materiału wideo znajduje się na tyle wysoko, że swoim zasięgiem może obejmować również fragment sąsiedniego pasa,

trawnika lub krawężnika, które znajdują się obok drogi. Są to zdecydowanie obszary infrastruktury miejskiej, gdzie obiekty podlegające analizie nie będą się znajdować. W związku z tym, można okroić szerokość obrazu. Jednak z długością również można postąpić w podobny sposób. Aplikacja jest zainteresowana jedynie niewielkim odcinkiem drogi, po którym poruszają się pojazdy. Można założyć, że dystans ok. trzech metrów wystarczy do odpowiedniego określenia prędkości, więc długość klatki filmowej również może zostać zmniejszona. W przypadku testowego materiału wideo, rozmiar przetwarzanego obszaru może zostać aż o połowę, co w ogromnym stopniu będzie rzutowało na wydajność algorytmów.

Można również pokusić się o zmianę rozdzielczości nagranych filmów. Jednak należy liczyć się z faktem, że takie posunięcie może doprowadzić do zmniejszenia dokładności wyników produkowanych przez program. Kamery, których lokalizacja spełnia wymagania aplikacji, czyli odpowiednie, zenitalne położenie względem ulicy znajdują się przeważnie na tyle wysoko, że niewielkie zmiany rozdzielczości klatki filmowej powinny wpływać w minimalnym stopniu. W związku z tym, należy empirycznie dobrać wystarczającą jakość obrazu. Zdecydowanie, aby rezultaty były satysfakcjonujące rozdzielczość może być mniejsza niż w przypadku testowych materiałów wideo (640 na 480 pikseli).

10 Ocena możliwości wdrożenia aplikacji

Główną zaletą stworzonej aplikacji jest jej prostota oraz przydatność dla społeczeństwa. Jednak warto się zastanowić czy wdrożenie jej do życia miejskiego jest możliwe. Należy wziąć pod uwagę dwa główne czynniki, od których to będzie zależeć, czyli dostępna infrastruktura monitoringu miejskiego oraz możliwości sprzętowe i kadrowe, którymi dysponuje urząd miasta.

Pierwszy z punktów, czyli zbiór kamer wideo jest właściwie wszędzie na świecie sukcesywnie poszerzany i na dzień dzisiejszy ciężko jest znaleźć miasto, w którym ludzie nie byliby obserwowani bez przerwy przez sprzęt nagrywający. Co więcej, w większości polskich miast pojawiają się kamery znajdujące się bezpośrednio nad ulicą, co jest warunkiem koniecznym dla prawidłowego działania programu. Na dzień dzisiejszy można stwierdzić, że infrastruktura jest dobrze dopasowana do wdrożenia aplikacji, a z biegiem czasu sytuacja ta będzie się poprawiać.

Z drugiej strony, większość urzędów miejskich może nie dysponować odpowiednimi możliwościami sprzętowymi, które pozwoliłyby na przeprowadzenie analizy natężenia ruchu drogowego. Należy liczyć się z koniecznością występowania co najmniej jednego procesora na kamerę wideo, której obraz ma zostać przetworzony. Na szczęście ceny

sprzętu komputerowego nie są wysokie i można z pewnością założyć, że wiele miast byłoby stać na zakup kilku tysięcy maszyn, które miałyby analizować ruch na ulicach. Przeszkolenie osób, które pracowałyby na stanowisku operatora aplikacji jest sprawą bardzo prostą. Interfejs użytkownika jest intuicyjny i przejrzysty, w związku z czym sprawa kadrowa nie stanowi przeszkody przy wdrożeniu aplikacji.

11 Perspektywy i ocena możliwości dalszego rozwoju

Istnieje wiele możliwości dalszego rozwoju aplikacji. Obecnie cała uwaga została skupiona na analizie ruchu drogowego. Naturalnym następstwem aktualnych funkcjonalności programu byłoby wyłączenie fotoradarów, które często znajdują się przy drogach. Koszt takiego urządzenia to ok. 100 000 zł plus dodatkowe koszty utrzymania, czyli serwisowanie i utrzymywanie łączy internetowych. W dzisiejszych czasach miasta są wyposażone w bogatą infrastrukturę monitoringu wideo, więc urzędy mogłyby zaoszczędzić bardzo dużą ilość pieniędzy na zaprzestaniu montowania fotoradarów, a zamiast tego skupiając się na wdrożeniu aplikacji. Dodatkowo, stworzony program dzięki wykorzystaniu analizy obrazu, mógłby sprawdzać poprawność jazdy kierowców nie tylko pod względem prędkości, ale również inne wykroczenia drogowe jak np. zmiana pasa na linii ciągłej, wjazd pod zakaz, przejazd na czerwonym świetle, wyprzedzenie na przejściu dla pieszych i wiele innych. Aby połączyć wykroczenia drogowe z odpowiednim pojazdem, należałoby poświęcić uwagę zagadnieniu ALPR, czyli Automatic License-Plate Recognition. Jest to system automatycznego rozpoznawania tablic rejestracyjnych. Oczywiście twórcy biblioteki OpenCV przewidzieli zainteresowanie tą sprawą i zaproponowali gotowe implementacje pomagające w odczytywaniu tablic.

Kolejnym potencjalnym kierunkiem rozwoju aplikacji byłoby podjęcie tematu zliczania osób widocznych na zdjęciu lub filmie. Byłaby to nieoceniona pomoc przy okazji zbiorowych imprez lub podczas ewakuacji budynku w sytuacjach zagrożenia. Można znaleźć wiele zastosowań, gdzie dokładna informacja o liczbie ludzi byłaby bezcenna.

Warto również zastanowić się nad wykorzystaniem danych statystycznych przechowywanych w bazie danych na temat natężenia ruchu na danych ulicach. Mogłoby to pomóc przy planowaniu remontów drogowych lub przy organizacji imprez masowych.

Bibliografia

- [1] Picardi M., Background subtraction techniques: a review, The ARC Centre of Excellence for Autonomous Systems (CAS) Faculty of Engineering, UTS, April 15, 2004
- [2] Sen-Ching S., Cheung C., Robust techniques for background subtraction in urban traffic video, Center for Applied Scientific Computing
- [3] <http://opencv.org/about.html> dostęp (2017.11.02)
- [4] <https://www.java.com/pl/about/> dostęp (2017.11.02)
- [5] <https://www.postgresql.org/about> dostęp (2017.11.02)
- [6] <https://www.google.com/maps/about/> dostęp (2017.11.02)
- [7] https://www.tomtom.com/pl_pl/drive/tomtom-traffic/ dostęp (2017.11.02)
- [8] <http://www.targeo.pl/> dostęp (2017.11.02)
- [9] <http://helion.pl/ksiazki/thinking-in-java-edycja-polska-wydanie-iv-bruce-eckel,thij4v.htm> dostęp (2017.11.02)
- [10] <http://helion.pl/ksiazki/bazy-danych-i-postgresql-od-podstaw-richard-stones-neil-matthew,bdpspo.htm> dostęp (2017.11.02)
- [11] <http://shop.oreilly.com/product/9780596516130.do> dostęp (2017.11.02)
- [12] <http://helion.pl/ksiazki/czysty-kod-podrecznik-dobrego-programisty-robert-c-martin,czykov.htm> dostęp (2017.11.02)
- [13] <http://helion.pl/ksiazki/algorytmy-struktury-danych-i-techniki-programowania-wydanie-iv-piotr-wroblewski,algo4.htm> dostęp (2017.11.02)
- [14] <http://helion.pl/ksiazki/programowanie-wspolbiezne-systemy-czasu-rzeczywistego-pawel-majdzik,provsp.htm> dostęp (2017.11.02)
- [15] <http://helion.pl/ksiazki/algorytmy-i-struktury-danych-alfred-v-ahojohn-e-hopcroft-jeffrey-d-ullman,alstrd.htm> dostęp (2017.11.02)
- [16] <http://helion.pl/ksiazki/jezyk-uml-2-0-w-modelowaniu-systemow-informatycznych-stanislaw-wrycza-bartosz-marcinkowski-krzysztof,juml2.htm> dostęp (2017.11.02)
- [17] Zgierski M., Aplikacja wykorzystująca obraz z kamery do analizy natężenia ruchu drogowego FTIMS PŁ 2017

USING THE CAMERA IMAGE TO ANALYZE THE TRAFFIC FLOW

Summary - The article describes the original application that uses the camera image to analyze the traffic flow. The image-based application is able to estimate the average speed of moving objects. The application works in real time has low hardware requirements. The conclusions from its operation as well as implementation and development perspectives were presented.

Keywords: traffic intensity, image analysis