Warszawa 2017/18

Filip Szatkowski
grupa 33ip
Prowadzący: Daniel Koguciuk

POLITECHNIKA WARSZAWSKA
WYDZIAŁ MECHATRONIKI
INSTYTUT AUTOMATYKI I ROBOTYKI

WIDZENIE MASZYNOWE

Rozpoznawanie oraz śledzenie konkretnej osoby na filmie na podstawie jej zdjęcia

1. Temat

Celem projektu było stworzenie programu umożliwiającego identyfikację oraz śledzenie konkretnej osoby na filmie na podstawie jej zdjęcia.

Założenia:

- osoba poszukiwana na pewno znajduje się na filmie przez cały czas, oraz przynajmniej przez jakąś jego część patrzy prosto w kamerę
- mała ilość ludzi na filmie(brak tłumów)
- dobre oświetlenie

Do testów wybrano fragmenty znanych filmów/teledysków ze względu na dużą dostępność zdjęć występujących w nich aktorów/muzyków.

2. Opis wykorzystanych technik

Klasyfikator pozwala uzyskać informację o tym, czy dany fragment obrazu zawiera szukany obiekt. Klasyfikator uczony jest za pomocą pozytywnych oraz negatywnych przykładów przeskalowanych do takiego samego rozmiaru. Następnie przesuwa się go po obrazie i porównuje z fragmentami obrazu. Klasyfikator zwraca informację o tym, czy dany region może zawierać szukany obiekt. Procedura jest wykonywana kilkakrotnie dla różnych rozmiarów klasyfikatora, który jest zaprojektowany tak, aby był łatwo skalowalny.

Klasyfikator kaskadowy składa się z wielu prostszych klasyfikatorów(etapów), które są po kolei aplikowane do danego fragmentu obrazu aż do etapu, gdzie wszystkie zostaną odrzucone lub obiekt przejdzie wszystkie testy. Klasyfikatory na każdym etapie są zbudowane z podstawowych klasyfikatorów za pomocą techniki boostingu. Użyty klasyfikator korzysta z cech Haara – proste kształty takie jak krawędzie, linie itp. są wykrywane poprzez przykładanie odpowiednich masek do obrazu.

EMD - Earth's Mover Distance(metryka Wassersteina) jest to dystans między dwoma rozkładami prawdopodobieństwa(w przypadku tego projektu – dwa normalizowane histogramy). Rozkłady interpretowane są jako dwie sterty piasku, EMD to natomiast najmniejszy koszt zamienienia jednej sterty w drugą, gdzie koszt to ilość piasku razy odległość. W przypadku tego projektu, za pomocą EMD sprawdzano podobieństwo obrazów na podstawie ich histogramów.

3. Działanie programu

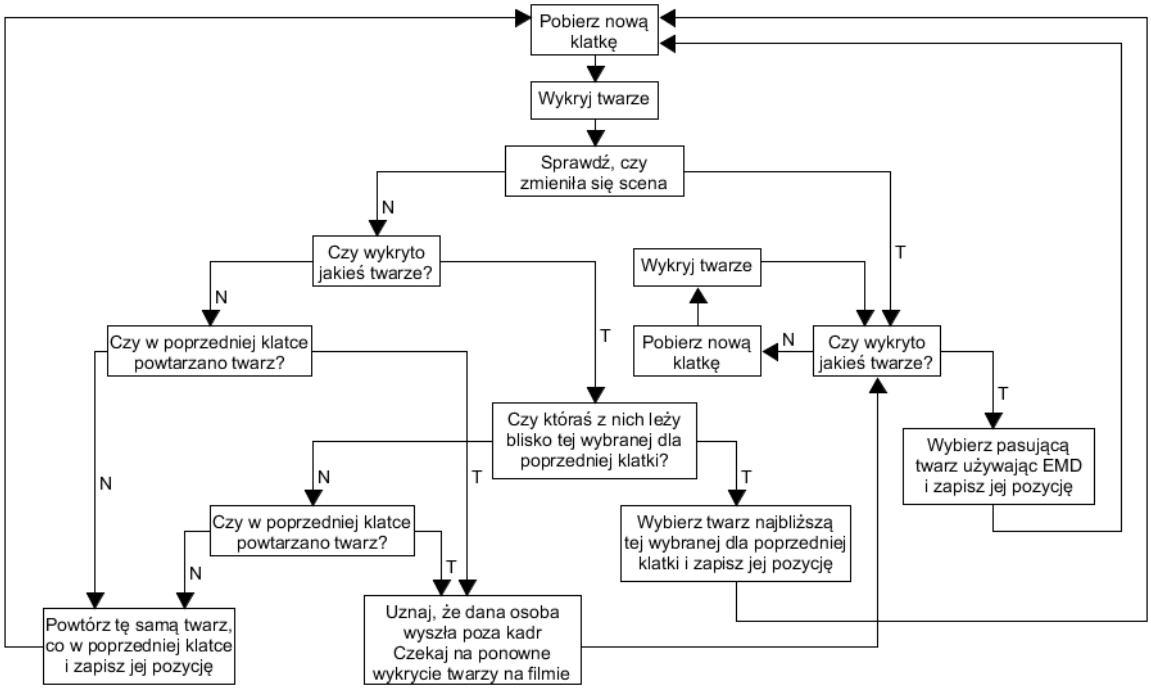
Program wczytuje zdjęcie i tworzy klasyfikator kaskadowy (**CascadeClassifier**) do wykrywania twarzy za pomocą pliku **haarcascade_frontalface_alt.xml**, który zawiera gotowy nauczony klasyfikator.

Następnie za pomocą funkcji **cvtColor** tworzona jest czarno-biała wersja wczytanego zdjęcia. Za pomocą metody **detectMultiScale** klasyfikatora wykrywana jest na zdjęciu twarz. Metoda zapisuje wykryte twarze jako wektor prostokątów będących ich obrysem; parametry metody określają, jak skalowane są maski przykładane w poszczególnych etapach(**scale**), ile cech musi być wykryte, aby przejść do kolejnego etapu klasyfikatora(**minNeighboors**) oraz jaka jest minimalna wielkość wykrytej twarzy(**minSize**). Aby uzyskać dla szukanej twarzy jak najlepsze dopasowanie, przyjęto scale = 1.05, minNeighboors = 3 i minSize = Size(20,20). Następnie otrzymany za pomocą klasyfikatora prostokąt wycinany jest ze źródłowego zdjęcia.

Na potrzeby późniejszego porównywania twarzy w tym miejscu wyliczany jest histogram obrazu. Funkcja **hsv_hist** przyjmuje obraz źródłowy w BGR, współrzędne oraz wielkość twarzy jako argumenty, wycina z obrazu twarz i przeskalowuje ją do rozmiaru twarzy szukanej(algorytm EMD wymaga takich samych rozmiarów obrazów) oraz konwertuje obraz do przestrzeni barw HSV. Następnie tworzony jest histogram kanałów H(hue – odcień) i S(saturation – nasycenie) dla odpowiednio 30. oraz 32. przedziałów. Jest on jeszcze normalizowany i zwracany jako macierz.

Po przygotowaniu wczytanego zdjęcia wczytywany jest film, na którym szukamy danej osoby. W pętli wczytywane oraz obrabiane są kolejne klatki filmu, które na potrzeby klasyfikatora konwertowane są do czarno-białego obrazu zapisywanego w oddzielnej macierzy. Aby przyspieszyć działanie klasyfikatora, poszczególne klatki są trzykrotnie zmniejszane, a klasyfikator jest nieco mniej dokładny – w parametrach funkcji **detectMultiScale scale** zwiększone jest do 1.2, natomiast **minSize** zwiększone do (30, 30). Wykryte prostokąty są odpowiednio korygowane – ich współrzędne oraz rozmiary są trzykrotnie zwiększane.

Pasująca twarz jest rysowana na klatce i wyświetlana. Decyzja o tym, która twarz spośród wykrytych pasuje do wczytanego zdjęcia, jest podejmowana za pomocą EMD z uwzględnieniem tego, czy zmienił się kadr oraz pozycji twarzy z poprzedniej klatki.



Rys. 1. Schemat podejmowania decyzji o wyświetlaniu twarzy

Jeśli kadr się zmienił(lub dla pierwszej klatki), czekamy na wykrycie przez klasyfikator twarzy. Po wykryciu jednej lub kilku twarzy decyzję o tym, która jest tą szukaną podejmujemy porównując je po kolei z twarzą wzorcową(EMD). Dalej jeśli scena na filmie się nie zmieni stosowane jest śledzenie wybranej twarzy – algorytm EMD jest kosztowny obliczeniowo i wykonywanie go na każdej klatce uniemożliwiłoby przetwarzanie obrazu w czasie rzeczywistym. Jeśli na danej klatce wykryte zostaną twarze, ich pozycja jest porównywana z pozycją twarzy wybranej w poprzedniej klatce. Jeśli któraś z nich jest dostatecznie blisko tej wybranej na poprzedniej klatce, uznawana jest za prawidłową. Jeśli nie znaleziono żadnej twarzy o współrzędnych podobnych do tej wybranej na poprzedniej klatce, przez jedną klatkę jest wyświetlana ta sama twarz, co klatkę wcześniej. Tak samo postępujemy w wypadku nie wykrycia żadnych twarzy na klatce. Jeśli poprzednia klatka miała uzupełnianą twarz oraz w następnej klatce nie wykryto żadnych twarzy, uznajemy, że szukana osoba przemieściła się poza kadr bądź obróciła twarz i ponownie czekamy na wykrycie twarzy, przeprowadzamy na nich EMD itd.

Funkcja **emd** jest zaimplementowana w OpenCV i jako parametry przyjmuje dwie sygnatury oraz rozdaj metryki wybrany do obliczenia odległości tych sygnatur(w projekcie użyto metryki euklidesowej). Sygnatury histogramów tworzone są przez przypisanie każdej kombinacji przedziałów hue i saturation trzech wartości: hue, saturation(odpowiadających współrzędnym x, y) oraz ilości występowania takiej kombinacji na obrazie.

O zmianie sceny decyduje się przez porównanie poprzedniej oraz aktualnej klatki – funkcją **absdiff** oblicza się ich różnicę, na tej różnicy przeprowadza się thresholding binarny i funkcją **sum** oblicza się sumę wartości pixeli tej różnicy. Jeśli przekracza ona wartość 30*wielkość obrazu uznaje się, że nastąpiła zmiana sceny.

4. Ograniczenia w działaniu programu

Oczywistym ograniczeniem programu jest wymaganie, aby osoba, której szukamy, cały czas była obecna na filmie i patrzyła prosto w kamerę lub, jeśli jest nieobecna, aby na filmie nie było żadnych innych osób. Dzieje się tak dlatego, że program wykonuje przez większość czasu śledzenie jednej twarzy, a wybór twarzy następuje tylko w chwili, gdy zmienia się scena. Ponadto jeśli osoba, której szukamy, obraca się, wychodzi poza kadr, zasłania na chwilę swoją twarz, klasyfikator kaskadowy może przestać wykrywać jej twarz, co skutkuje błędami w działaniu programu.

Takie działanie jest w znacznym stopniu wymuszone przez wybrany sposób porównywania twarzy – na podstawie uzyskanej za pomocą algorytmu EMD odległości można jedynie zdecydować, która twarz na danej klatce jest najbliższa szukanej, niemożliwe jest natomiast odrzucenie którejś z twarzy, ponieważ odległości te mogą być różne w zależności od oświetlenia, jakości filmu i orientacji twarzy.

Dość znacznym ograniczeniem jest także koszt wykrywania oraz porównywania twarzy – wykrywanie twarzy z większą dokładnością lub częstsze porównywanie znalezionych twarzy prowadziło do znaczącego spowolnienia działania programu, który nie mógł już działać w czasie rzeczywistym. Wskazuje to na problem przy ewentualnym zastosowaniu programu do filmów, na których znajdują się większe ilości ludzi i następowałoby przetwarzanie dużej ilości twarzy.

5. Rozwój programu

W celu usprawnienia działania programu należałoby przede wszystkim zastosować inny sposób porównywania twarzy, który pozwoliłby odrzucać twarze niewystarczająco podobne do szukanej postaci.

Klasyczne metody porównywania dwóch twarzy jako dwóch obrazów nie są w stanie zwrócić informacji o tym, czy dana twarz może być tą szukaną, czy nie. Dobrym podejściem mogłoby być wytrenowanie własnego klasyfikatora za pomocą wczytywanego zdjęcia, chociaż do uzyskania dobrych rezultatów prawdopodobnie potrzebne byłoby więcej danych uczących(zdjęć). Taki klasyfikator zastąpiłby etap wykrywania twarzy i pozwolił na ominięcie fazy porównywania zdjęć. Wykonywanie operacji klasyfikacji na każdej klatce, analogicznie do wykrywania twarzy na każdej klatce w programie, powinno być możliwe do realizacji w czasie rzeczywistym – wtedy śledzenie twarzy byłoby wykorzystywane tylko do korekcji ewentualnych błędów klasyfikatora. Program korzystający z takiego klasyfikatora mógłby także działać w czasie rzeczywistym dla filmów z większą ilością osób dzięki ominięciu fazy porównywania twarzy, która jest najbardziej kosztowna obliczeniowo.

Innym możliwym podejściem jest zastosowanie bardziej zaawansowanych algorytmów śledzenia twarzy, możliwe jednak, że byłoby to zbyt kosztowne obliczeniowo i uniemożliwiałoby pracę w czasie rzeczywistym. Poza tym nie rozwiązuje to problemu decyzji o tym, którą twarz należy śledzić. Uodporniłoby to jednak program na sytuacje takie jak obrót śledzonej osoby.

W miarę prostym usprawnieniem mechanizmu śledzenia byłoby dodanie do algorytmu wnioskowania informacji o więcej niż jednej poprzedniej pozycji wybranej twarzy, wprowadza to jednak trochę problemów – jeśli na przykład dana osoba się porusza, należałoby to uwzględnić.

Kolejną możliwością jest wprowadzenie do algorytmu decydującego o wyborze twarzy wnioskowania na podstawie następnych kilku klatek. Mogłoby to nieco usprawnić śledzenie twarzy, a widz raczej nie zauważyłby opóźnienia rzędu paru klatek. W tym przypadku należałoby się zastanowić nad konsekwencjami wprowadzenia opóźnienia, które może być w niektórych zastosowaniach niepożądane lub wręcz groźne (np. szukamy terrorysty w tłumie).

Jeśli nie jest istotne działanie w czasie rzeczywistym, a należy np. prześledzić nagranie z monitoringu, możliwe byłoby przypisywanie twarzy po przetworzeniu całego filmu, dzięki czemu zastosować można zarówno wnioskowanie o danej klatce na podstawie przeszłych oraz przyszłych klatek, jak i zastosowanie bardziej kosztownych obliczeniowo metod (na przykład użycie kilku klasyfikatorów zamiast jednego, ustawienie bardziej dokładnych parametrów klasyfikatora).