Politechnika Krakowska im. Tadeusza Kościuszki

Projekt zrealizowany w ramach
Informatycznego Obozu Przetrwania

„IntelligentEye”

Autorzy:
Mateusz Chrząszcz
Anna Czarnota
Mateusz Gaweł
Paweł Góralik
Bartosz Górecki
Mateusz Gruszka
Bartłomej Grzebinoga
Michał Łoza
Łukasz Obszyński

Opiekun:
dr inż. Piotr Kowalski

2012-09-27

Spis treści

1 Założenia projektu

Celem projektu Inteligenteye jest wykonanie programu wykrywającego podejrzane zachowania na podstawie analizy obrazu z kamery wideo.

1.1 Technika wykonania

Projekt będzie stworzony w języku JAVA. Do rozpoznawania obrazu będzie użyta biblioteka OpenCV, opakowana za pomocą JavaCV.

i cos z reszty zespolow

1.2 Analiza obrazu

Obraz otrzymywany z kamery komputerowej będzie analizowany pod kątem przede wszystkim rozpoznawania twarzy. Następnie zostanie przeszukana baza danych zawierająca portrety osób podejrzanych. W momencie wychwycenia podobieństwa program poinformuje użytkownika o niebezpieczeństwie. W miarę możliwości program będzie rozpoznawał również wszelkiego rodzaju podejrzane pakunki i przedmioty a także oznaki wskazujące na nerwowość i zaniepokojenie obserwowanych osób.

2 Wstęp teoretyczny

2.1 Język JAVA

Java jest obiektowym językiem programowania, powstałym w roku 1995. Poprzez standardowe jak i rozszerzone biblioteki wkracza w różnorodne rejony zastosowań takie jak np. karty inteligentne i elektronika, systemy zarządzania bazami danych, obsługa multimediów, internet, grafika 3D, kryptografia, itd. Co więcej JAVA jest niespotykanie bezpiecznym środowiskiem i umożliwia w znaczny sposób kontrolę i sterowanie bezpieczeństwem. Zdecydowanie różni się od innych języków trzeciej generacji tym, że jest językiem interpretowanym a nie kompilowanym. Oznacza to, że powstały w wyniku kompilacji kod wynikowy nie jest programem jaki można niezależnie uruchomić lecz stanowi tzw. Beta-kod, który jest interpretowany przez Maszynę Wirtualną (JavaVM) pracującą w określonym środowisku. Ze względu na kod nie istotne jest na jakim sprzęcie będzie uruchamiana aplikacja. Ważna jest tylko Maszyna Wirtualna. Jest to niezwykle ciekawy pomysł umożliwiający odcięcie się od wszystkich poziomów sprzętowo-programowych będących poniżej Maszyny Wirtualnej. Koncepcja ta jest powszechna również w samym języku JAVA, dzięki czemu poprzez stworzenie abstrakcyjnych klas i metod podstawowe biblioteki Javy nie muszą być nieustannie rozbudowywane.

2.2 Rozpoznawanie obrazu

Elementami składowymi kompletnego rozpoznawania obrazów są trzy poziomy: niskiego, średniego i wysokiego poziomu. Przetwarzanie niskiego poziomu obejmuje odbiór obrazu, przetwarzanie wstępne, oraz poprawę jakości obrazu (jak np. eliminacja zakłóceń, zmiana kontrastu czy filtracja). Przetwarzanie średniego poziomu polega na na segmentacji i wydzielaniu obiektów obrazu. Wysoki poziom odpowiada za klasyfikację, rozpoznanie i interpretację analizowanej sceny.

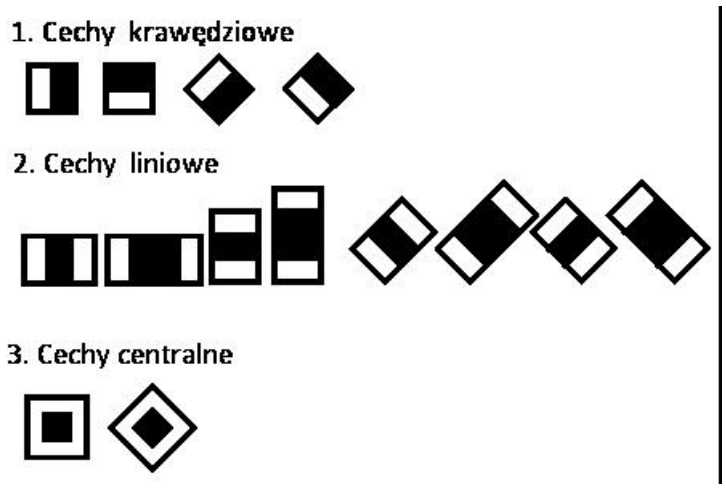
3 Rozpoznanie twarzy

Rozpoznanie twarzy zostało zrealizowane za pomocą otwartej biblioteki OpenCV opakowanej przez JavaCV w celu umożliwienia jej użycia przez język Java. Udostępnia ona szereg narzędzi

ułatwiających rozpoznanie obrazu i jego dalszą obróbkę. Aby algorytmy poprawnie rozpoznawały twarze, został wygenerowany plik posiadający listę cech charakterystycznych dla twarzy. Generacja pliku polega na podaniu zestawu zdjęć "pozytywnych", czyli zawierających tylko twarze z frontu i profilu, zdjęć "negatywnych" ukazujących jedynie tła i środowisko w których twarze nie występują, oraz zwykłych zdjęć ludzi w zwykłym otoczeniu. Aby rozpoznać cechy typowe dla twarzy stosowane są algorytmy Haara i Viola-Jones'a (rozwiniętym przez Rainera Lienhart'a oraz Johena Maydt'a).

3.1 Wybór zestawu cech

W metodzie Viola-Jenes cechy traktowane są jako kombinacja dwóch, trzech lub czterech prostokątów. Ilość możliwych kombinacji połączenia prostokątów była bardzo duża, dlatego przy określonych założeniach została wyselekcjonowana grupa podstawowych cech. Następnie udoskonalony zestaw cech jeszcze lepiej identyfikujący poszukiwane obiekty został zaproponowany w metodzie Lienhart-Maydt. Grupa ta została podzielona na cechy krawędziowe, liniowe oraz centralnie otoczone. Właśnie z tego zestawu cech korzysta funkcja z OpenCV. Zasada działania cech polega na odnajdywaniu obszarów, na których różnica pomiędzy sumą pikseli regionów ograniczonych czarnymi i białymi prostokątami znajduję się powyżej pewnego progu. Na podstawie tej wartości algorytm może zidentyfikować czy badany obszar jest poszukiwanym obiektem czy jego tłem.



Rysunek 1: Zestaw cech zaproponowany przez Lienhart'a i Maydt'a

Dla przykładu, prostokątne regiony oczu mają dużo mniejszą intensywnośd niż obszar czoła, ten zaś dużo większą od obszaru włosów

3.2 Tworzenie klasyfikatora kaskadowego

Odbywa się z wykorzystaniem algorytmu uczącego AdBoost. Wykorzystuje on tzw. wzmocnienie adaptacyjne (ang. adaptive boosting). Jego zadaniem jest wygenerowanie silnego klasyfikatoraz kaskady słabych. W przypadku obrazów, na których wykryty ma zostać obiekt wykorzystuje się technikę przesuwnego okna. Z tak zeskanowanego obrazu generowany jest zbiór okien, które poddawane są weryfikacji przez klasyfikatory. Na podstawie informacji z tych okien słabe klasyfikatory uczą się odrzucać obszary, na których na pewno nie znajduje się poszukiwany obiekt. W każdej pętli algorytm skupia się na źle wykrytych obszarach przez poprzednie klasyfikatory. Nadaje im wyższe wagi (gdyż w następnej iteracji istnieje większe prawdopodobieostwo prawidłowego wykrycia obiektu). Wykorzystywanie tych wag przez następne klasyfikatory pozwala na zawężenie przestrzeni poszukiwń do coraz mniejszego obszaru. Tak skonstruowany silny klasyfikator umożliwia szybkie wykrywanie poszukiwanych obiektów.

4 Porównanie twarzy

Twarze rozpoznane na zdjęciu zostają przekazane w celu ich porównania z twarzami dostępnymi w bazie. Porównywanie twarzy opiera się o szereg algorytmów.

4.1 Kowariancja

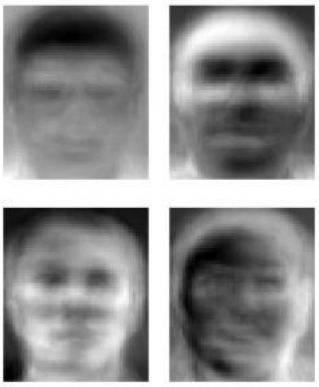
Kowariancja jest miarą tego, jak dalece wartości dwóch zmiennych losowych zmieniają się razem. Jeśli zmienne zachowują się podobnie, to kowariancja jest liczbą dodatnia, w przeciwnym razie kowariancja jest ujemna. Macierz kowariancji to macierz której elementem na miejscu (i,j) jest kowariancja pomiędzy i-tym oraz j-tym elementem wektora losowego.

4.2 Analiza głównych składowych (PCA)

Statystyczna metoda służąca odnajdywaniu struktur w zbiorze zmiennych losowych. Celem PCA jest przekształcenie zbioru obserwacji prawdopodobnie skorelowanych ze sobą zmiennych w zbiór wartości nieskorelowanych zmiennych zwanych głównymi składowymi. Transformacja ta jest zdefiniowana w taki sposób, że pierwsza główna składowa ma jak największą wariancję (czyli wyjaśnia tak dużą część zmienności danych jak to tylko możliwe), a każdy kolejny element ma jak największą wariancję pod warunkiem, że jest nie jest skorelowany z poprzednią składową.

4.3 Eigenface

Eigenface, lub twarze własne, to zbiór wektorów własnych używany przy komputerowym rozpoznaniu twarzy. Zbiór twarzy własnych może zostać wygenerowany przez wykonanie procedury PCA na dużym zbiorze obrazów ludzkich twarzy. Twarze własne można traktować jak zbiór "standaryzowanych składników twarzy". Każda ludzka twarz może być pojmowana jako kombinacja tych standardowych twarzy. Przykładowo czyjaś twarz może być kompozycją twarzy uśrednionej plus 10% cech z twarzy własnej 1, 1,55% z twarzy własnej nr 2 oraz nawet 3% z twarzy własnej 3.



Rysunek 2: Przykłady twarzy własnych z laboratorium AT&T w Cambridge

Dzięki przechowywaniu twarzy nie jako pliku graficznego, a jako listy pewnych wartości, rozmiar pliku jest niewielki. Stworzone twarze własne będą ukazywać się jako jasne ciemne obszary układające się w pewien określony zbiór.

4.4 Algorytm k najbliższych sąsiadów

Jest to algorytm uzywany w statystyce do prognozowania wartości pewnej zmiennej losowej, jednak może być również używany do klasyfikacji. Aby polepszyć działanie algorytmu kNN

powszechnie stosowaną techniką jest standaryzacja lub normalizacja danych. Standaryzacja polega na doprowadzeniu do sytuacji w której wartość średnia poszczególnej cechy ma wartość 0 a odchylenie standardowe = 1. Normalizacja to doprowadzenie do sytuacji w której wartości zmiennej należą do przedziału [0,1]

Uczenie:

1. Dokonaj alternatywnie: standaryzacji/normalizacji/pozostaw dane jakie są
2. Zapamiętaj cały zbiór treningowy

Testowanie:

1. Dokonaj standaryzacji/normalizacji/pozostaw dane jakie są (testowanie)
2. Policz odległości pomiędzy wektorem testowym a wszystkimi wektorami zbioru treningowego
3. Posortuj odległości od największej do najmniejszej
4. Zobacz etykiety k – najbliższych wektorów do wektora testowego. Zrób histogram częstości poszczególnych etykiet spośród „k-najbliższych" (Policz ile i których etykiet było spośród k najbliższych)
5. Przypisz najczęściej występującą etykietę jako etykietę wektora testowego
6. Jeśli wystąpił impas (dwie klasy miały taką samą liczbę głosów) rozwiąż problem losowo

5 Praktyczna realizacja zadania

6 Wnioski i uwagi

Spis rysunków