

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

SYSTÉM PRE VYHODNOCOVANIE EMOCIONÁLNEHO STAVU
POUŽÍVATEĽA POMOCOU WEB KAMERY

DIPLOMOVÁ PRÁCA

2017

Bc. FILIP PRIKLER

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

SYSTÉM PRE VYHODNOCOVANIE EMOCIONÁLNEHO STAVU
POUŽÍVATEĽA POMOCOU WEB KAMERY

DIPLOMOVÁ PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra informatiky
Školiteľ: PaedDr. Martin Magdin Ph.D.

Nitra 2017

Bc. FILIP PRIKLER



Univerzita Konštantína Filozofa v Nitre
Fakulta prírodných vied

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Filip Prikler
Študijný program: Aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.9 aplikovaná informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Systém pre vyhodnocovanie emocionálneho stavu používateľa pomocou web kamery

Anotácia: Zisťovanie emocionálneho stavu používateľa patrí v súčasnosti medzi základné oblasti umelej inteligencie, v ktorej sa využívajú prvky neurónových sietí.

Ciele diplomovej práce: Študent navrhne a zrealizuje systém, ktorým bude možné určiť v reálnom čase emocionálny stav používateľa na základe web kamery

Požiadavky na vedomosti a zručnosti študenta: Dobré znalosti v oblasti využívania jazyka PHP, JAVA a C, prípadne neurónových sietí.

Školiteľ: PaedDr. Martin Magdín, Ph.D.
Oponent: doc. PaedDr. Jozef Kapusta, PhD.
Katedra: KI - Katedra informatiky

Dátum zadania: 02.10.2015

Dátum schválenia: 14.10.2015


prof. Ing. Milan Turčáni, CSc.
schválil/a

POĎAKOVANIE

Chcem sa úprimne poďakovať svoju školiteľovi PaedDr. Martinovi Magdinovi, Ph.D. za veľa hodnotných konzultácií, inšpiráciu, za odbornú pomoc, cenné rady a pripomienky pri vypracovaní diplomovej práce. Rád by som poďakoval tiež mojej rodine a snúbenici za silnú podporu.

ABSTRAKT

PRIKLER, Filip: Systém pre vyhodnocovanie emocionálneho stavu používateľa pomocou web kamery [Diplomová práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied. Školiteľ: PaedDr. Martin Magdin, Ph.D. Stupeň odbornej kvalifikácie: Magister odboru Aplikovaná informatika. Nitra : FPV, 2017. 50 s.

Diplomová práca sa zaoberá problematikou vyhodnocovania emocionálneho stavu používateľa pomocou webkamery. Hlavným cieľom je implementácia systému, ktorý bude schopný automaticky klasifikovať emócie používateľa v reálnom čase na základe vstupných dát z webkamery. V teoretickej časti je uvedený prehľad historických prístupov výskumníkov z danej oblasti. V tejto časti sú taktiež popísané metódy na detekciu tváre, extrakciu črt tváre a klasifikáciu emócií. Samostatné kapitoly sú venované knižnici OpenCV a Viola-Jones detektoru. V praktickej časti je opísaný postup tvorenia dvoch riešení – detekcie tváre v jazyku Java a klasifikácia emócií pomocou SDK od spoločnosti Affectiva. Klasifikovali sme 6 základných emocionálnych výrazov podľa Ekmana: radosť, smútok, hnev, znechutenie, prekvapenie a strach.

Kľúčové slová: Detekcia tváre. Extrakcia črt. Klasifikácia emócií. OpenCV. SDK Affectiva. Emocionálny výraz

ABSTRACT (v anglickom jazyku)

PRIKLER, Filip: System for evaluation of the emotional state of user by web camera [Diploma Thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences. Supervisor: PaedDr. Martin Magdin, Ph.D. Degree of Qualification: Master of applied informatics. Nitra : FNS, 2017. 50 p.

The diploma thesis deals with the evaluation of the user's emotional state using a webcam. The main goal is to implement a system that will be able to automatically classify user emotions in real time based on webcam input data. In the theoretical part, there is an overview of the historical approaches of researchers from the given area. This section describes methods of face detection, facial extraction, and emotional classification. Separate chapters are dedicated to the OpenCV library and the Viola-Jones detector. In the practical part the steps of creating two solutions - face detection in Java and ethical classification thanks to Affectiva SDK are described. We have classified six basic emotional expressions according to Ekman: joy, sadness, anger, disgust, surprise, and fear.

Keywords: Face Detection. features extraction. Emotion classification. OpenCV. SDK Affectiva. Emotion expression

OBSAH

ÚVOD	11
1 ANALÝZA SÚČASNÉHO STAVU	12
1.1 Stručná história výskumu rozpoznávania emócií.....	12
1.2 FILTWAM.....	14
1.3 FaceTracker.....	17
1.4 Databázy tvárí / kolekcie.....	20
2 CIELE DIPLOMOVEJ PRÁCE	22
3 TEORETICKÉ VÝCHODISKÁ PRÁCE	23
3.1 Metódy detekcie tváre v obraze	23
3.1.1 Znalostné metódy	24
3.1.2 Metódy založené na porovnávaní šablón	25
3.1.3 Metódy založené na vzhľade.....	25
3.1.4 Metódy invariantných rysov	26
3.2 Metódy extrakcie črt	26
3.2.1 Lineárna analýza	27
3.2.2 Elastické zhlukové grafy	30
3.2.3 Aktívny model vzhľadu (AAM)	31
3.3 Metódy klasifikácie emócií.....	32
3.3.1 Skrytý Markov Model	32
3.3.2 Bayesov klasifikátor.....	33
3.4 OpenCV	34
3.5 Viola-Jones detektor	35
4 NÁVRH A RIEŠENIE Č.1	38
4.1 Detekcia tváre v jazyku Java.....	38
4.2 Používateľské rozhranie.....	39
4.3 Realizácia samotného systému.....	39
4.4 Zhrnutie.....	41
5 RIEŠENIE Č.2, VÝSLEDKY A ICH ZHODNOTENIE	42
5.1 SDK od spoločnosti Affectiva	42
5.2 Databáza výrazov tváre od Affectiva.....	45
5.3 Používateľské rozhranie.....	47
5.4 Vývoj finálnej aplikácie	48
5.4.1 Popis tried a metód aplikácie	48

5.5	Zhodnotenie dosiahnutých výsledkov	54
5.6	Možnosti budúceho vývoja	57
ZÁVER.....		58
ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV		59
ZOZNAM PRÍLOH		63

ZOZNAM ILUSTRÁCIÍ, TABULIEK A UKÁŽOK

Obrázok 1	Schéma jednotlivých vrstiev FILTWAM-u (Bahreini, Nadolski, & Westera, 2014).....	15
Obrázok 2	Ilustrácia extrakcie črt tváre a stratégia pre optimalizáciu	17
Obrázok 3	Sada 26 bodov vybraných z originálnych 66, ktoré identifikoval FaceTracker (Loconsole, Chiaradia, Bevilacqua, & Frisoli, 2014).....	18
Obrázok 4	Ukážka sledovania tváre s čiastočným zakrytím	19
Obrázok 5	Ukážka detekcie tváre pomocou znalostných metód v praxi (Ming-Hsuan, Kriegman, & Ahuja, 2002)	25
Obrázok 6	Šablóna určená pre lokalizáciu tváre založená na metóde Sinha (Scassellati, 1998).....	25
Obrázok 7	Princíp metódy založenej na vzhľade.....	26
Obrázok 8	Proces lineárnej analýzy (Delac, Grcic, & Grcic, Independent comparative study of PCA, ICA and LDA on the FERET data set., 2005).....	28
Obrázok 9	a) zmiešané body pri premietnutí na priamku b) oddelené body pri premietnutí na inú priamku (Delac & Grgic, Face Recognition., 2007)	30
Obrázok 10	Elastický porovnávací diagram aplikovaný na obrázok tváre (Wiskott, Fellous, & Krüger, 1997).....	30
Obrázok 11	Príklad Gáborovej vlnkovej transformácie (Loos, Wiecek, & Würtz1, dátum neznámy)	31
Obrázok 12	Aktívny model vzhľadu a) inicializácia b) pozícia po prvej iterácii c)pozícia po druhej iterácii d) konvergované (Cootes, 1998)	31
Obrázok 13	Príznaky podobné Haar kaskáde (OpenCV, 2017).....	37
Obrázok 14	Používateľské rozhranie aplikácie na detekciu tváre.....	39
Obrázok 15	Vyťaženie systémových zdrojov počas detekcie tváre.....	42
Obrázok 16	Reťazec automatizovaného spracovania tváre (McDuff, et al., 2016)	44
Obrázok 17	Prehľad princípu fungovania webového frameworku na zbieranie tvárových dát z verejnosti (McDuff, a iní, 2013)	46
Obrázok 18	Príklad porovnania medzi predikciami klasifikátora (zelená línia) a ručne zadanými popismi (modrá a čierna prerušovaná línia) pre 6 videí vybraných z databázy (McDuff, a iní, 2013).....	47
Obrázok 19	Ukážka používateľského rozhrania aplikácie	48

Obrázok 20	Diagram triedy DrawingCanvas.cs.....	51
Obrázok 21	Vyťaženie systémových zdrojov počas klasifikácie emócií.....	54
Obrázok 22	Ukážka určovania miery úspešnosti	55
Obrázok 23	Ukážka klasifikácie hnevu.....	56
Obrázok 24	Detekcia šťastia a tváre z obrázku.....	57
Tabuľka 1	Podmnožina antropometrických bodov na tvári, ktoré sú použité na výpočet geometrie črt tváre (Loconsole, Chiaradia, Bevilacqua, & Frisoli, 2014).....	18
Tabuľka 2	Prehľad databáz obsahujúci fotografie tvárí ukazujúcich rôzne emócie.	20
Tabuľka 3	Miera úspešnosti rozpoznávania emócií.....	56
Ukážka 1	Knižnice použité na implementáciu detekcie tváre	40
Ukážka 2	Dôležité objekty systému pre detekciu tváre	40
Ukážka 3	Proces detekcie tváre a vykresľovanie obdĺžnikov pre nájdené tváre	41
Ukážka 4	Trieda FilePath.cs	48
Ukážka 5	Metóda Window_Loaded()	49
Ukážka 6	Metóda StartCameraProcessing()	51
Ukážka 7	Zbieranie dát pre vyhodnotenie štatistických údajov	52
Ukážka 8	Renderovanie klasifikácie emócie	54

ÚVOD

Problematika vyhodnocovania ľudských emócií pomocou počítača je veľmi zaujímavá téma, ktorá si za posledné roky získava čoraz viac pozornosti. Za povšimnutie určite stojí nevšedné prelínanie odborov informatiky a psychológie. Má veľký potenciál zlepšiť interakciu medzi človekom a počítačom v rôznych oblastiach – od edukačnej sféry, cez medicínu až po oblasť komercie.

V prvej kapitole sa nachádza prehľad výskumov v oblasti vyhodnocovania emócií tváre od roku 1649. V tejto kapitole sú uvedené rôzne prístupy a metódy, ktoré mali najväčší vplyv na rozvoj danej oblasti.

V druhej kapitole sú stanovené hlavné a čiastkové ciele diplomovej práce, ktoré budú prácou naplnené.

Tretia kapitola predstavuje teoretické východiská práce. Čitateľ má možnosť sa v nej zoznámiť s metódami, ktoré je možné aplikovať v rámci detekcie tváre v obraze, s extrakciou črt a samotnou klasifikáciou prežívaných emócií. V tejto kapitole je tiež opísaná knižnica OpenCV (z angličtiny „Open Computer Vision“), ktorá na open-source platforme poskytuje verejnosti silný nástroj na spracovanie obrazu počítačom. Samostatnú podkapitolu je venovaná aj robustnému algoritmu Viola-Jones detektoru, ktorý má veľké výhody pri spracovaní obrazu – je veľmi málo náchylný na zmenu svetelných podmienok a poskytuje vysokú spoľahlivosť.

Štvrtá kapitola sa zameriava na opis nami navrhnutého prvého riešenia detekcie tváre v jazyku Java. V tejto kapitole sa nachádza zdokumentovaný postup pri vypracovaní úlohy, návrh používateľského rozhrania ako aj realizácia vo vývojovom prostredí.

V poslednej kapitole je uvedené finálne riešenie problematiky vyhodnocovania emocionálneho stavu používateľa pomocou webkamery prostredníctvom „Software Development Kit“ od spoločnosti Affectiva. V tejto kapitole je uvedený princíp fungovania daného riešenia, ako aj spôsob získania unikátnej databázy, pomocou ktorej sú emócie klasifikované. Ďalej je opísaný postup pri práci s SDK. Čitateľovi môže poslúžiť ako návod a uľahčiť mu tak prácu. V kapitole sú popísané najdôležitejšie triedy a metódy, ktoré boli použité na klasifikáciu emócií v reálnom čase a priebežné zbieranie štatistických údajov. V závere kapitoly je uvedené komplexné zhodnotenie dosiahnutých výsledkov.

1 ANALÝZA SÚČASNÉHO STAVU

Táto kapitola predstavuje prierez výskumov zameraných na rozpoznávanie emócií zo statického obrazu alebo videosekvencie, ktoré boli realizované v minulosti a ovplyvnili aj súčasný vývoj problematiky. V analýze súčasného stavu sa čitateľ taktiež zoznámí so systémom FILTWAM, ktorý sa používa v sfére e-learningu a taktiež so systémom FaceTracker.

1.1 Stručná história výskumu rozpoznávania emócií

Výskum zaoberajúci sa emóciami tváre sa začal už v 17. storočí a stal sa základom pre ďalší rozvoj v tejto oblasti. V knihe od Johna Bulwera s názvom "Pathomyotomia", ktorá bola publikovaná v roku 1649, je podrobne rozobraná problematika výrazu tváre a pohybov svalov na tvári. (Magdin, Turčáni, & Hudec, 2016)

Od začiatku roku 1970, Paul Ekman a jeho kolegovia vykonali rozsiahle štúdie výrazu ľudskej tváre (Ekman, Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique., 1994). Zistili, že existujú dôkazy, ktoré podporujú univerzálnosť vo výraze tváre. Tieto "univerzálne výrazy" sú nasledovné:

- šťastie
- smútok
- hnev
- strach
- prekvapenie
- znechutenie.

Paul Ekman a jeho kolegovia študovali mimiku v rôznych kultúrach vrátane predliterárnych kultúr, a našli mnoho zhôd vo výraze a rozpoznávaní emócií na tvári. Vo svojej štúdii však zároveň pozorovali aj rozdiely vo výrazoch, a navrhli model, v ktorom sa výrazy tváre riadia „pravidlami vystupovania na verejnosti“ v rôznych sociálnych kontextoch. Napríklad japonské subjekty výskumu a americké subjekty ukazovali podobné výrazy tváre pri pozeraní na rovnaký stimulačný film. Avšak japonskí diváci boli menej ochotní ukázať svoje skutočné výrazy za prítomnosti úradníkov. Na druhú stranu sa zdá, že

deti majú širokú škálu výrazov tváre bez toho aby sa ich učili, čo naznačuje, že tieto výrazy sú vrodené.

Ekman a Friesen v roku 1978 vyvinuli „Facial Action Coding System“ (FACS) na kódovanie výrazov tváre, kde sú pohyby na tvári popísané súborom akčných jednotiek (AU) (Ekman & Friesen, Facial Action Coding System: Investigator's Guide, 1978). Každá AU jednotka je založená na príbuznosti svalov. Tento systém kódovania mimiky sa vykonáva ručne pomocou sady predpísaných pravidiel. Ako vstupy slúžia statické obrázky mimiky, často predstavujúce vrchol výrazu. Tento proces je časovo veľmi náročný.

Ekmanova práca inšpirovala mnoho výskumníkov (L. Davisa, T. Otsuku, J. Ohya, Y. Yacooba a ďalších), aby analyzovali výrazy tváre pomocou spracovania obrazu a videa. Sledovaním tvárových črt a meraním množstva pohybu na tvári sa pokúsil roztriediť rôzne výrazy tváre do jednotlivých kategórií. Práca Tian (Y.Tian, T.Kanade, & J.Cohn, 2001) zameraná na analýzu a rozpoznávanie výrazov tváre používa tieto "základné výrazy" alebo ich podmnožinu. Vo svojej práci Pantic a Rothkrantz poskytujú hĺbkovú rekapituláciu mnohých z výskumov automatického rozpoznávania výrazu tváre uskutočnených v posledných rokoch (Pantic & Rothkrantz, 2000).

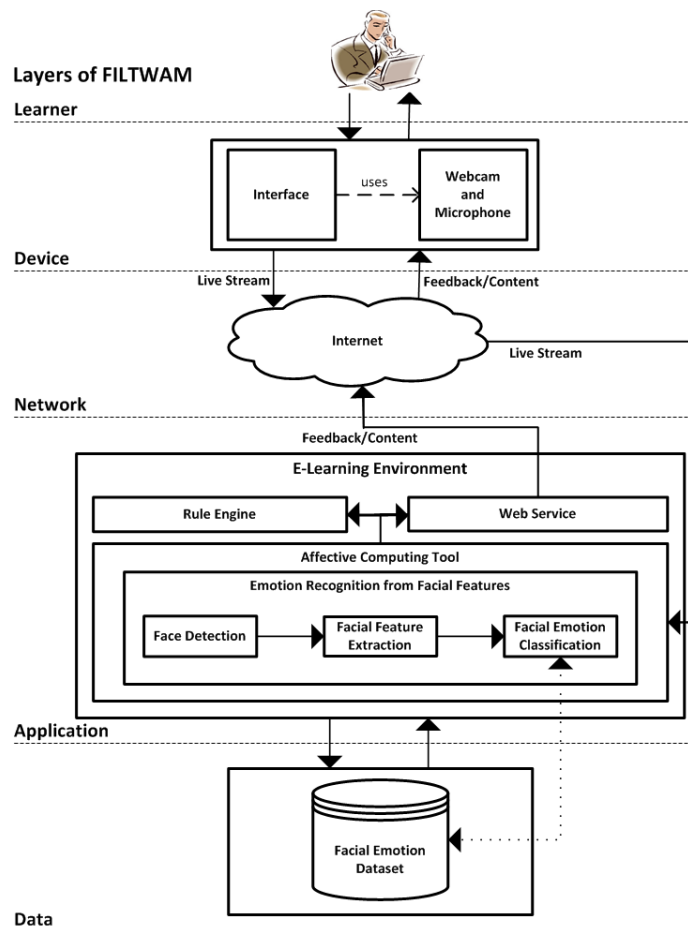
Výskum v oblasti kvantifikácie výrazov tváre za pomoci počítačov sa nezačal až do deväťdesiatych rokov dvadsiateho storočia. Mase použil optický tok za účelom rozpoznania výrazu tváre (Mase, 1991). Bol jedným z prvých, ktorý použil postupy spracovania obrazu na rozpoznanie mimiky. Lanitis a kol. použili model flexibilného tvaru a model vzhľadu pre kódovanie obrazu, identifikáciu osôb, určenie telesnej pózy, rozpoznanie pohlavia a rozpoznanie výrazu tváre (Lanitis, Taylor, & Cootes, 1995). Black a Yacoob extrahovali lokálne parametrizované modely z pohybu obrazu za účelom stabilizácie pohybu (Black & Yacoob, 1995). Akonáhle boli získané, tieto parametre boli použité ako vstupy do klasifikátora založeného na pravidlách, ktoré sú schopné rozpoznať šesť základných výrazov tváre. Yacoob a Davis vypočítali optický tok a použili podobné pravidlá pre klasifikáciu šiestich výrazov tváre (Yacoob & Davis, 1996). Rosenblum, Yacoob a Davis tiež vypočítali optický tok oblastí na tvári a potom za účelom klasifikácie výrazov aplikovali sieť založenú na radiálnych funkciách (Rosenblum, Yacoob, & Davis, 1996). Essa a Pentland použili metódu optického toku jednotlivých oblastí tváre na rozpoznanie výrazu (Essa & Pentland, 1997). Donato a kol. testovali rôzne funkcie pre rozpoznanie tvárových AU a týmto spôsobom klasifikovali výraz tváre na snímke (Donato, Bartlett, Hager, Ekman, & Sejnowski, 1999). Otsuka a Ohya najprv vypočítali optický tok,

potom vypočítal 2D Furiérove transformačné koeficienty, ktoré boli použité ako vektory črt pre „Hidden Markov Model“ (HMM) na dosiahnutie klasifikácie výrazu (Otsuka & Ohya, Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences, 1997). Trénovaný systém bol schopný rozpoznať jednu zo šiestich výrazov takmer v reálnom čase (asi 10 Hz). Okrem toho používali sledované pohyby na ovládanie výrazu tváre animovaného systému Kabuki (Otsuka & Ohya, A study of transformation of facial expressions based on expression recognition from temporal image sequences, 1997). Martinez predstavil prístup indexovania založeného na identifikácii obrazu s frontálne natočenou tvárou ku snímaciemu zariadeniu za rôznych svetelných podmienok a výrazov tváre (Martinez, 1999). Bayesianov prístup bol použitý za účelom nájdenia najlepšej zhody medzi lokálnymi pozorovaniami a naučeným modelom lokálnych črt. „Hidden Markov Model“ bol použitý na dosiahnutie dobrého rozpoznania výrazu, aj keď nové podmienky nezodpovedali podmienkam, ktoré sa vyskytli vo fáze učenia. Oliver a kol. použili sledovanie spodnej časti tváre pre extrahovanie vlastností tvaru úst a použili ich ako vstupy do systému klasifikácie emócií založenom na „Hidden Markovom Modeli“ (systém rozpoznával neutrálny výraz, radosť, smútok a otvorené ústa) (Oliver, Pentland, & Lafter:, 2000).

Tieto metódy sú podobné v tom, že najprv extrahujú niektoré črty z obrázkov, potom tieto prvky sa používajú ako vstupy do klasifikačného systému, a výsledok je jedna z vopred vybraných emočných kategórií. Líšia sa hlavne spôsobom extrakcie črt z videozáznamov a jednotlivými klasifikátormi použitými na rozlišovanie medzi rôznymi emóciami (Prikler, 2016).

1.2 FILTWAM

Uvedený systém na vyhodnocovanie emócií v reálnom čase sa používa v sfére e-learningu a jeho autorom je Kiavash Bahreini. Skratka FILTWAM (Framework for Improving Learning Through Webcams And Microphones) znamená „Framework pre zlepšovanie vyučovania pomocou webkamery a mikrofónu“ a ponúka aktuálnu a relevantnú spätnú väzbu založenú na výraze tváre a verbálnych prejavoch študenta.



Obrázok 1 Schéma jednotlivých vrstiev FILTWAM-u (Bahreini, Nadolski, & Westera, 2014)

Framework FILTWAM obsahuje päť funkčných vrstiev a niekoľko čiastkových komponentov vnútri jednotlivých vrstiev. Sú to tieto vrstvy:

- 1) študent
- 2) zariadenie
- 3) sieť
- 4) aplikácia
- 5) dáta.

Obrázok 1 ilustruje jednotlivé prepojenia v rámci frameworku FILTWAM.

Vrstva študenta – táto vrstva sa týka subjektu, ktorý využíva webové výučbové materiály v rámci rozvoja svojej osobnosti alebo pri príprave na skúšku.

Vrstva zariadení zachycuje pracovné miesto študenta, či už sa jedná o osobný počítač, notebook alebo inteligentné zariadenia, ktoré v sebe obsahujú webovú kameru a mikrofón pre zber užívateľských dát.

Sieťová vrstva používa internet na vysielanie naživo prenášať žiaka a prijímať dáta v reálnom čase na žiaka.

Aplikačná vrstva je najdôležitejšou súčasťou FILTWAM-u. Skladá sa z e-learningového prostredia a niekoľkých čiastkových komponentov. E-learningové prostredie používa webovú kameru a technológiu rozpoznávania emócií z tváre s cieľom uľahčiť študentovi proces učenia. Obsahuje tri čiastkové zložky s názvom: nástroj pre výpočet emócií, komponent obsahujúci pravidlá a webový server. Aplikačná vrstva je srdcom FILTWAM-u. Spracováva dáta získané z tváre a hlasovú intonáciu žiaka. Skladá sa z komponentov pre rozpoznávanie emócií z tvárových črt a hlasovej intonácie. V našej práci sa obmedzíme na klasifikáciu emócií založenej na prúde dát z webkamery. V nasledovných sekciách si viac priblížime jednotlivé komponenty aplikačnej vrstvy.

Rozpoznávanie emócií z tvárových črt - tento komponent extrahuje črty z tváre a klasifikuje emócie. Zahŕňa v sebe tri komponenty, ktoré umožňujú rozpoznanie a kategorizáciu konkrétnej emócie.

Detekcia tváre - proces rozpoznávania emócií z tvárových črt sa začína na komponente detekcie tváre. Tento komponent nie je určený na to, aby rozpoznal tvár konkrétnej osoby, namiesto toho je naprogramovaná na odhalenie tváre a rozpoznanie emócií.

Extrakcia črt tváre - akonáhle je tvár detegovaná, komponent pre extrakciu črt nájde dostatočný počet bodov, ktoré definujú tvárové črty študenta. Tieto body sú považované za významné rysy študentovej tváre a môžu byť extrahované automaticky.

Klasifikácia emócie tváre - framework FILTWAM používa známe prístupy ku klasifikácii emócií, ktoré boli často používané počas posledných tridsiatich rokov. Použité prístupy sa zameriavajú na klasifikáciu šiestich základných emócií podľa Ekmana a Friesena. Komponent klasifikujúci emócie vo frameworku FILTWAM podporuje týchto šiestich základných emócií a neutrálnu emóciu, ale v prípade potreby dokáže rozpoznať iné emócie či jemnejšie výrazy tváre. Tento komponent analyzuje video sekvencie a môže v rámci analýzy extrahovať obrázok pre každý snímok videa. Klasifikačný komponent je nezávislý na rase, veku, pohlaví, účese, okuliaroch, pozadia, alebo fúzoch.

Komponent obsahujúci pravidlá - komponent spravuje didaktické pravidlá a spúšťa príslušné pravidlá pre poskytovanie spätnej väzby, a slúži aj na prispôsobenie učebného materiálu študentovi prostredníctvom zariadenia. Opísaný komponent určený pre elektronické vzdelávanie vyhovuje špecifickým pravidlám didaktického prístupu k vyučovaniu študentov.

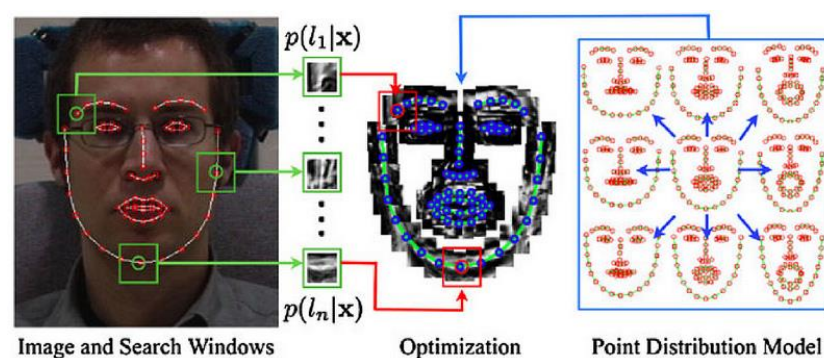
Webová služba - komponent webová služba prenáša spätnú väzbu a učebný materiál študentovi. V tejto fáze študent môže získať spätnú väzbu na základe jeho / jej emócií.

Dátová vrstva je fyzické úložisko klasifikovaných emócií. Zahŕňa v sebe sadu dát tvárových emócií, ktorá tvorí intelligenčný kapitál systému. Dátová vrstva zároveň zaznamenáva štatistickú referenciu pre detekciu emócií. (Bahreini, Nadolski, & Westera, 2014)

1.3 FaceTracker

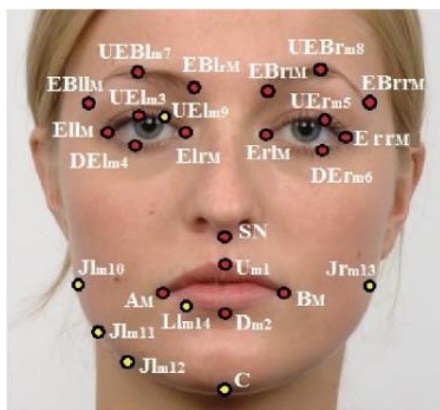
FaceTracker je knižnica vytvorená za účelom sledovania pohybov tváre a napísaná v jazyku C++ za použitia knižnice OpenCV 3. Jej autorom je Jason Saragih, ktorý ju vytvoril v roku 2010 a aktuálne ju udržiava Kyle McDonald. Knižnica je dostupná zdarma pre nekomerčné účely.

Prístup tejto knižnice redukuje nepresnosti detekcie tváre, prináša nízku online výpočtovú zložitosť a vysokú efektivitu detekcie. Systém od Saragih najprv pomocou Haar kaskád určí, kde sa na obrázku môže nachádzať tvár. Nasledujúci krok je extrakcia črt tváre (Obrázok 2).



Obrázok 2 Ilustrácia extrakcie črt tváre a stratégia pre optimalizáciu

FaceTracker identifikuje a lokalizuje 66 orientačných bodov na tvári v 2D priestore. Claudio Loconsole a spol. v článku „Real-Time Emotion Recognition: An Improved Hybrid Approach for Classification Performance“ vybrali a opísali sadu 26 bodov.



Obrázok 3 Sada 26 bodov vybraných z originálnych 66, ktoré identifikoval FaceTracker (Loconsole, Chiaradia, Bevilacqua, & Frisoli, 2014)

Excentrické črty tváre sú určené vypočítaním excentrickosti elipsy vytvorenej za pomoci špeciálnych bodov na tvári. Z hľadiska geometrie sa excentrickosť elipsy vypočíta ako rozdielnosť elipsy oproti svojmu kruhovému tvaru. Pre elipsu je hodnota excentrickosti väčšia alebo menšia ako nula. V prípade hodnoty rovnkej nule sa jedná o kruh. Ako príklad môžeme uviesť vykresľovanie elipsy na základe bodov ohraničujúcich ústa – pokiaľ sa pozorovaná tvár usmieva, hodnota je väčšia ako nula, ale pri prekvapení sa hodnota blíži k nule. Podobný fenomén sa dá pozorovať v oblasti obočí a očí.

Tabuľka 1 Podmnožina antropometrických bodov na tvári, ktoré sú použité na výpočet geometrie črt tváre (Loconsole, Chiaradia, Bevilacqua, & Frisoli, 2014)

Číslo	Bod	Označenie	Oblasť
1	Pravý kútik	Am	Ústa
2	Ľavý kútik	Bm	Ústa
3	Horná pera	Um1	Ústa
4	Dolná pera	Dm2	Ústa
5	Ľavý kútik	ElIm	Ľavé oko
6	Pravý kútik	Elrm	Ľavé oko
7	Horné viečko	DEIm3	Ľavé oko
8	Dolné viečko	DEIm4	Ľavé oko
9	Ľavý kútik	ErIm	Pravé oko
10	Pravý kútik	Errm	Pravé oko
11	Horné viečko	UErm5	Pravé oko
12	Dolné viečko	DErm6	Pravé oko
13	Koniec obočia	EblIm	Ľavé obočie
14	Vnútorne nadobočie	Eblrm	Ľavé obočie
15	Nadobočie	UEBIm7	Ľavé obočie
16	Vnútorne nadobočie	EbrIm	Pravé obočie
17	Koniec obočia	Ebrrm	Pravé obočie
18	Nadobočie	UEBrm8	Pravé obočie
19	Analaktický bod	SN	Nos

20	Horné viečko	UElm9	Ľavé oko
21	Ľavý kranio-metrický bod	JLm10	Čeľusť
22	Ľavá horná sánka	Jlm11	Čeľusť
23	Ľavá dolná sánka	Jlm12	Čeľusť
24	Kranio-metrický bod	C	Brada
25	Pravý kranio-metrický bod	Jrm13	Čeľusť
26	Ľavá dolná pera	Llm14	Ústa

Po dokončení identifikácie a lokalizácie bodov na tvári je vytvorený model tváre. Následne je spustený algoritmus, ktorý doladí pozíciu bodov podľa optimalizačnej šablóny. Nie je možné meniť pozíciu jednotlivých bodov modelu nezávisle na sebe. Ako skvelý príklad v tomto prípade poslúžia body obočia, ktoré na seba svojou pozíciou nadväzujú. Vo všeobecnosti, všetky body jednej črty tváre v systéme FaceTracker spolu súvisia. (Borenstein, 2012)



Obrázok 4 Ukážka sledovania tváre s čiastočným zakrytím

Obrázok 4 znázorňuje sekvenciu 4 rovnakých obrázkov a použitie rôznych algoritmov na optimalizovanie pozície bodov na tvári pri čiastočnom zakrytí. V tomto prípade sa jedná o obzvlášť náročnú situáciu, keďže noviny nie len zakrývajú oči, ale tlmia aj svetlo dopadajúce na tvár. V poradí od hornej sekvencie k dole boli použité nasledovné algoritmy:

1. ML(Maximum Likelihood) prispôbenie s Gaussovým jadrom
2. MAP (Maximum a posteriori Estimation) prispôbenie s Gaussovým jadrom
3. ML prispôbenie s German-McClurovým jadrom

4. MAP prispôsobenie s German-McClurovým jadrom.

Obrázok 4 vyjadruje, že ML a MAP varianty CLM prispôsobenia s Gaussovým jadrom zlyhali. ML prispôsobenie s German-McClurovým jadrom vykazuje lepšiu robustnosť keďže účinky zakrytia v oblasti očí nie sú tak výrazné. Avšak nahromadenie chýb viedlo k strate sledovania tváre ku koncu sledovanej sekvencie. Výsledok MAP prispôsobenia s German-McClurovým jadrom bolo v tomto prípade najúspešnejšie, keďže sa mu podarilo sledovať tvár až do konca sekvencie. (Saragih, Lucey, & Cohn, 2010)

1.4 Databázy tváří / kolekcie

Za posledné roky sa vyvinulo množstvo metód a algoritmov pre detekciu a identifikáciu tváre a preto bol dôraz na vývin databáz, či už pre tréning klasifikátorov alebo pre testovanie. Niektoré databázy sú zamerané napríklad na variabilitu osvetlenia, pozadia alebo natočenie tváre. Ich obsah pozostáva väčšinou z fotografií vo farbách alebo v odtieňoch šedej. Tabuľka 2 poskytuje stručný prehľad voľne dostupných databáz.

Tabuľka 2 Prehľad databáz obsahujúci fotografie tváří ukazujúcich rôzne emócie.

Názov databázy	Počet fotografií	Rozlíšenie [px]	Farebnosť
BioID face database	1521	384x286	nie
Faces94	< 4000	180x200	áno
Faces 1999 (Front)	450	596x592	áno
The Yale Face Database B	5760	640x480	nie
CBCCL Face Database #1	31022	19x19	nie
The extended Yale Face Databaset	16128	168x192	nie
Georgia Tech face database	750	640x180	áno
The ORL face database	400	92x112	nie
BioID face database	1521	384x286	nie

BioID face database - databáza obsahuje 1521 šedo-tónových čelných obrazov tváre 23 rôznych osôb. Každý snímok je v rozlíšení 384x286 pixelov. Pri vytváraní databázy sa kládol dôraz na reálne podmienky. Z toho vyplýva, že ponúka širokú variabilitu osvetlenia, pozadia a veľkosti tváre. Jednotlivé obrázky sú uložené vo formáte PGM. (BioID, 2017)

Faces 1999 (Front) - databáza bola vytvorená Markusom Weberom na California Institute of Technology. A obsahuje priame fotografie tváre. Nájdeme v nej 450 tvárí 27 rôznych ľudí. Fotografie sú odfotené na rozličnom pozadí, s rôznym osvetlením a mimikou. Formát obrázkov je JPEG v rozlíšení 596x592 pixelov.

The extended Yale Face Databaset - databáza rozširuje existujúcu The Yale Face Database B a pozostáva z 16128 fotografií 28 rôznych ľudí pod 9 rozličnými pózami. Jednotlivé obrazy vznikli pod 64 rôznymi svetelnými podmienkami. Rozlíšenie každej fotografie je 168x192 pixelov.

CBCL FACE DATABASE #1 obsahuje čiernobiele fotografie v rozlíšení 19x19 pixelov. Pozostáva z celkovo 2901 fotografií obsahujúcich tvár (faces) a 28 121 neobsahujúcich tvár (nonfaces). Tieto fotografie sú rozdelené na trénovaciu a testovaciu sadu. Formát fotografií je pgm.

Georgia Tech face database je databázou fotografií 50-tich ľudí, ktoré boli získané pri rôznych svetelných podmienkach a s rôznou mimikou tváre. Rozlíšenie sa v priemere pohybuje okolo 150x150 pixelov a každá fotografia je farebná, uložená vo formáte jpeg.

The ORL Database of Faces - táto databáza obsahuje čiernobiele fotografie 40-tich ľudí, od každého človeka po 10 vzoriek. Všetky snímky boli získané pri homogénnom pozadí, no s rozličnými svetelnými podmienkami a s rozličnou mimikou tváre. Taktiež obsahuje niekoľko tvárí s okuliarmi. Formát je pgm, veľkosť každého snímku je 92x112 pixelov. (Grgic & Delac, 2005)

2 CIELE DIPLOMOVEJ PRÁCE

Hlavným cieľom diplomovej práce je návrh a realizácia systému pre rozpoznávanie a vyhodnocovanie ľudských emócií v reálnom čase na základe výrazu tváre, získaného za pomoci webovej kamery. Výsledná aplikácia bude poskytovať možnosť analyzovať zozbierané údaje.

Čiastkové ciele sú:

- Opísať históriu vývoja výskumu v oblasti rozpoznávania emócií
- Opísať súčasné systémy, ktoré je možné použiť na detekciu a sledovanie tváre a klasifikáciu emočného stavu používateľa
- Opísať jednotlivé metódy slúžiace na detekciu tváre v obraze, extrakciu črt tváre a klasifikáciu emócií a na základe analýzy súčasného stavu navrhnúť vlastný systém.

V rámci stanovenia výskumnej otázky sme sa zamerali na porovnanie percentuálnej miery úspešnosti vyhodnotenia emocionálneho stavu ľudskej tváre pomocou nášho riešenia a porovnať dosiahnutý výsledok oproti hodnotám stanovenými napr. psychológmi (fotografie zo špeciálnych databáz).

Nami stanovená výskumná otázka znie: *„Aká je percentuálna úspešnosť vyhodnocovania emócií pomocou nami navrhnutého softvéru v porovnaní s jednoznačne určenými emocionálnymi stavmi osôb nachádzajúcich sa napríklad na fotografiách.“*

3 TEORETICKÉ VÝCHODISKÁ PRÁCE

V tejto kapitole sa zameriame na opísanie jednotlivých metód, ktoré sa dajú využiť pri tvorbe nášho systému. Princípy analýzy emocionálnych stavov vyjadrovaných prostredníctvom výrazov v tvári majú spoločný základ s princípmi používanými pri rozpoznávaní tváre. Celý proces analýzy sa dá rozdeliť do troch po sebe nasledujúcich fáz (Fasel & Luetin, 2003). Ide o detekciu a lokalizáciu tváre v obraze s komplexným pozadím, vrátane jeho normalizácie, ďalej o extrakciu vhodných príznakov popisujúcich daný výraz v tvári a nakoniec určenia zodpovedajúceho výrazu na základe extrahovaných príznakov pomocou zvoleného klasifikátora. Voľba sledovaných príznakov (klasifikátora) je vždy závislá od použitého prístupu a požadovaných výstupoch.

3.1 Metódy detekcie tváre v obraze

Táto problematika je len jednou z mnohých oblastí, ktorou sa zaoberá počítačové videnie. Detekcia tváre môže byť využitá vo všetkých prístrojoch, ktoré sú vybavené snímacím zariadením. Vďaka rozmachu informačných technológií v modernej dobe možno tieto metódy využiť v počítačoch, mobilných telefónoch, tabletoch, vstavaných systémoch, ktoré sú vybavené snímacím zariadením, alebo ak nie je problém k nim nejaké zariadenie pre snímanie pripojiť. Detekcia tváre sa napríklad využíva pri autentifikácii pomocou snímky z kamery. Veľké využitie má aj u bezpečnostných zložiek, kde na základe zaznamenania páchateľa je jeho tvár porovnaná s databázou a na základe zhody je určená jeho identita. Aj pre využitie v robotike je táto problematika zaujímavá (Umožňuje napríklad humanoidnému robotovi udržiavať zrakový kontakt s človekom).

V súčasnosti existuje viacero metód detekcie ľudskej tváre. Detekcia tváre sa vykonáva na základe analýzy zdrojovej snímky (alebo sekvencií snímok pri videu). Tieto metódy je možné rozdeliť na základe spôsobu prístupu k problému do štyroch skupín (Yang & Huang, 1994).

- **Znalostné metódy** (Knowledge-based methods) - využívajú sady pravidiel, ktoré sú vytvorené na základe znalostí o typickej tvári. Pravidlá zachytávajú vzťahy medzi typickými črtami tváre.
- **Metódy založené na porovnávaní šablón** (Template matching methods) - na základe preddefinovaných vzorov funkcií (manuálne preddefinovaných, alebo parametrizovaných) popisujúcich tvár, alebo jej časti, je vypočítaná korelácia so vstupným obrazom. Na základe výsledku korelácie je detegovaná tvár.

- **Metódy založené na vzhľade** (Appearance-based methods) - z databázy tváří, ktorá obsahuje najrôznejšie možné črty tváre alebo jej pootočená a pohľady, je vytvorený model na základe ktorého sa vykonáva detekcia.
- **Metódy invariantných rysov** (Feature invariant approaches) - tieto metódy hľadajú typické črty tváre, ktoré sú nezávislé na natočení hlavy (uhla získavanej snímky) a na svetelných podmienkach.

Existuje niekoľko faktorov, ktoré sťažujú detekciu tváre (Ming-Hsuan, Kriegman, & Ahuja, 2002):

Pozícia kamery / tváre (snímacieho zariadenia) - Zdrojové dáta môžu byť zachytené z rôznych uhlov vzhľadom k tvári. Môže teda dôjsť k zakrytiu niektorých charakteristických črt tváre, (napr. viditeľnosti iba jedného oka pri snímaní z profilu), alebo úplne iné rozmiestnenie charakteristických prvkov (očí, nosa, úst, obočie) než pri snímaní z prednej časti tváre.

Prítomnosť alebo absencia významných prvkov v tvári - tento problém vytvárajú hlavne porasty v podobe ochlpenia (fúzy, brada) ale i okuliare, atď. Problém nie je len v prítomnosti týchto prvkov, ale aj v ich možných tvaroch, farbách, veľkostiach, atď.

Výraz v tvári - priamo ovplyvňuje jeho vzhľad. Napríklad zatvorenie očí.

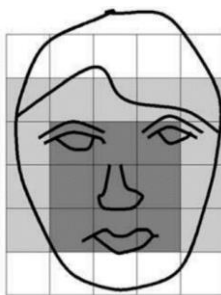
Zakrytie tváre - časť tváre môže byť zakrytá. Napríklad oko, vlasy alebo rôzne časti tváre môžu byť zakryté inými časťami tela pri skupinovej fotografii, prítomnosť ruky pred tvárou a podobne.

Svetelné podmienky – Problém spôsobuje intenzita svetla. Množstvo svetla ale i nedostatok svetla sa prejavuje šumom v obraze. Všetky tieto faktory výrazne ovplyvňujú výslednú snímku. Dochádza k zmenám typických farieb kože a môže tiež dôjsť k splynutiu niektorých charakteristických rysov tváre.

3.1.1 Znalostné metódy

Znalostné metódy sú založené na definovaných pravidlách, ktoré popisujú skutočnú ľudskú tvár. Pravidlá väčšinou definujú vzájomné vzťahy medzi jednotlivými časťami tváre. Tieto metódy sú navrhnuté prevažne na lokalizáciu tváre v obraze a sú pomerne náročné na implementáciu, využívajú zložité a robustné algoritmy. Z tohto dôvodu spravidla tieto metódy nedosahujú požadovaných výsledkov. Prvý krát metódu použili G. Yang and T. S. Huang za účelom detekcie tváre na zložitom pozadí. Publikovali o tom

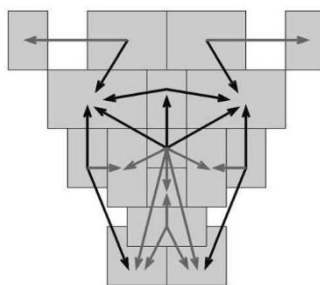
v roku 1994 článok s názvom “Human Face Detection in Complex Background”. Podrobnejšie je princíp znalostnej metódy popísaný v práci. (Yang & Huang, 1994)



Obrázok 5 Ukážka detekcie tváre pomocou znalostných metód v praxi (Ming-Hsuan, Kriegman, & Ahuja, 2002)

3.1.2 Metódy založené na porovnávaní šablón

Ako prvý predstavil uvedenú metódu Sinha. Metódy obsahujú databázu niekoľkých šablón, ktoré popisujú celú tvár, prípadne sú to šablóny obsahujúce jednotlivé časti tváre samostatne. Vyhľadávajú tvár na základe korelácie medzi vstupným obrazom a šablónami z databázy.

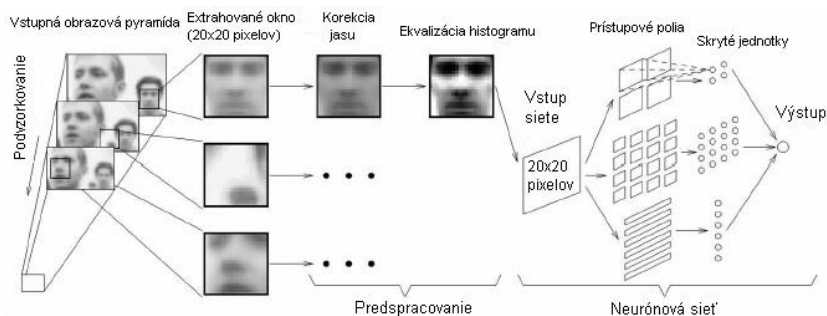


Obrázok 6 Šablóna určená pre lokalizáciu tváre založená na metóde Sinha (Scassellati, 1998)

Nevýhodou tejto metódy je, že aby sme docielili čo najlepšie výsledky detekcie, musíme mať čo najrozsiahlejšiu databázu, čo je pomerne prácne a aj časovo náročné. Tieto metódy, sú určené na lokalizáciu a aj detekciu tváre v obraze. Obrázok 3 predstavuje jednoduchý príklad znázorňujúci princíp metódy založenej na porovnávaní šablón. Pomer pixelov je 14x16. Šablóna pozostáva zo šestnástich oblastí (šedé obdĺžniky) a 23 vzťahov (znázornené šípkami). Metódu detekcie tváre založenej na porovnávaní šablón sa prvý krát pokúsili použiť T. Sakai, M. Nagao, a S. Fujibayashi, ktorý svoje zistenia opisujú v článku “Line Extraction and Pattern Detection in a Photograph” publikovanej v roku 1969. Podrobnejšie je tento prístup popísaný v knihe (Štanclová, 2009).

3.1.3 Metódy založené na vzhľade

Modely tváre sa získavajú učením z množiny obrázkov určených na tréning, ktorá obsahuje rôzne typy tvárí. Detekcia prebieha na základe porovnávania častí obrazu s týmto modelom. V tomto prípade je tiež nutné vytvoriť model tváre, i keď učenie prebieha poloautomaticky. Metódy založené na tomto princípe dosahujú veľmi dobré výsledky a sú určené prevažne na detekciu. Príklad detekčnej metódy založenej na vzhľade je uvedený na nasledujúcom obrázku (Obrázok 7), podrobný princíp je popísaný v práci (Sung & Poggio, 1998).



Obrázok 7 Princíp metódy založenej na vzhľade

3.1.4 Metódy invariantných rysov

Cieľom týchto metód je nájsť morfologické rysy, ktoré sa nemenia ani pri zmene natočenia tváre a rôznych svetelných podmienkach. Ako morfologické rysy sa využívajú napríklad tvárové príznaky (oči, ústa, atď.) alebo farba pokožky. Následne je za pomoci morfologických rysov lokalizovaná tvár v obraze. Výhodou týchto metód je jednoduchá a rýchla implementácia s pomerne kvalitnými výsledkami lokalizácie. Prvotný výskum v tejto oblasti bol publikovaný v roku 2003. Jedná sa o článok „A survey on Pixel-Based Skin Color Detection Technique“

3.2 Metódy extrakcie črt

Cieľom tejto metódy je zvoliť takú sadu črt, ktoré sú jedinečné pre každý špecifický výraz v tvári. Výber vhodných črt patrí v celej analýze k najdôležitejším, pretože iba zo správne zvolených črt tváre možno analyzovať zodpovedajúci emocionálny stav. Zvolené príznaky musia byť pre daný výraz tváre úplne nemenné pri rôznom pohlaví, veku, rase a vzhľade analyzovaného subjektu. Podľa (Pantic & Rothkrantz, 2000) môžeme rozlišovať dva základné typy tvárových črt:

Trvalé tvárové črty- sú vždy na tvári prítomné, ale môžu byť deformované na základe výskytu určitého výrazu v tvári. Sú to najmä oči, pery, obočie, ale aj permanentné vrásky či celková textúra tváre.

Prechodné tvárové črty - vrásky a vypukliny objavujúce sa len v prípade určitého výrazu v tvári. Najčastejšie na čele, v oblastiach okolo očí, kútikoch úst a podobne.

Podľa formátu vstupných obrazových dát (statické obrazy alebo videosekvencie) rozlišujeme extrakciu príznakov na:

Extrakciu črt deformácií - používa sa u statických obrazov aj u videosekvencií, črty sa extrahujú nezávisle z každého obrazu alebo snímky. Ide o črty, ktoré sú typické pre vyjadrenie rôznych deformácií jednotlivých častí ľudskej tváre (privreté oko, otvorené ústa a podobne), ktoré sa podieľajú na danom celkovom emocionálnom výraze.

Extrakciu črt pohybu - používa sa iba u video-sekvencií, príznaky sú extrahované na základe rozdielov pozícií (pohybu) sledovaných objektov v jednotlivých snímkach. Tieto črty slúžia na opis dynamiky jednotlivých častí ľudskej tváre (žmurkanie, pohyb úst a podobne), ktoré sa podieľajú na celkovom emocionálnom výraze. Keďže sa tieto črty menia s časom, musia byť tomuto prístupu prispôsobený aj výsledný klasifikátor.

Existuje mnoho rôznych metód a prístupov k extrakcii črt. Sú to napríklad metódy založené na lineárnej analýze. Z nich je možné menovať napríklad analýzu hlavných komponentov (PCA), analýzu nezávislých komponentov (ICA) a lineárne diskriminačnú analýzu (LDA). Niekedy lineárny prístup nestačí, tento nedostatok však riešia jadrové metódy, napríklad jadrová analýza hlavných komponentov (KPCA) a jadrová diskriminačná analýza (KDA). Iným prístupom môže byť metóda založená na elastických zhlukových grafoch (EBGM), ktorá sa snaží o kopírovanie topologickej štruktúry ľudskej tváre, či aktívny model vzhľadu (AAM), ktorý využíva štatistický model a mnoho ďalších.

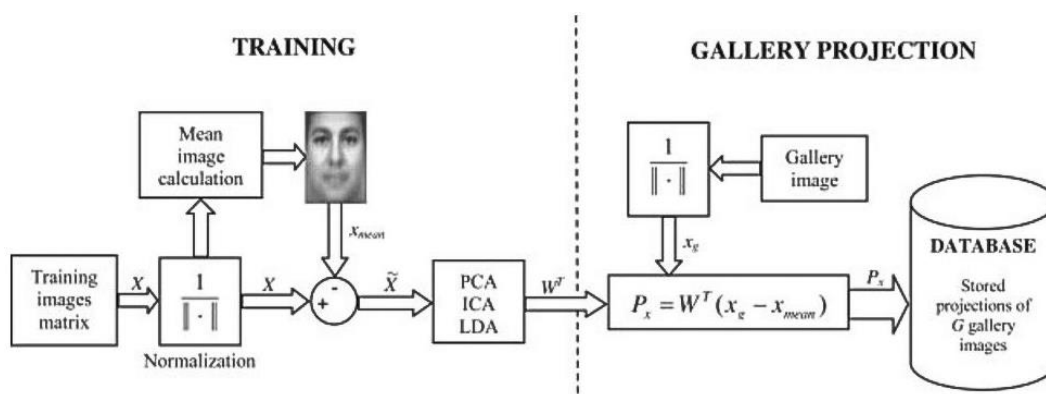
3.2.1 Lineárna analýza

Metódy lineárnej analýzy majú za úlohu lineárne transformovať pôvodné obrazové dáta reprezentované pomocou vektorov do projekčného vektora a tým súčasne nájsť vhodné lineárne rozdelenie priestoru, aby dáta jednotlivých tried boli čo najlepšie separovateľné. Medzi tri základné metódy lineárnej analýzy patria PCA, ICA a LDA.

Každá metóda má svoju vlastnú reprezentáciu mnohodoménového vektorového priestoru založenej na štatistických metódach.

Obrázok 8 znázorňuje princíp transformácie dát do projekčného vektora. V ľavej časti obrázku je zobrazené tréningovanie systému s redukciou priestoru a v pravej časti je možné vidieť proces projekcie pôvodných obrazových dát do priestoru s nižšou dimenziou. X je matica obsahujúca obrázky vyjadrené ako vektory v jednotlivých stĺpcoch matice, x_{mean} je priemerný obrázok (vo forme vektora), $\bar{X}$ je matica obsahujúca odčítané priemerné obrázky v stĺpcoch, W^T je projekčná matica získaná z lineárnej analýzy nad dátami $\bar{X}$ a x_g je obrázok z galérie reprezentovaný vektorom. Počas tréningovej fázy je vypočítaná projekčná matica W^T a potom ich obrazová galéria premietnutá do pod-priestoru $P_x = W^T (x_g - x_{mean})$, kde P_x je projekčný vektor, ktorý je následne uložený do databázy.

Pri procese identifikácie je najskôr vyššie uvedeným postupom vypočítaný projekčný vektor P_x a v nasledujúcom kroku je porovnávaný s projekčnými vektormi uloženými v databáze. Výsledkom je postupnosť vzdialeností, pričom výsledok je vybraný podľa najmenej vzdialenosti.



Obrázok 8 Proces lineárnej analýzy (Delac, Grcic, & Grcic, Independent comparative study of PCA, ICA and LDA on the FERET data set., 2005)

Analýza hlavných komponentov (Principal Component Analysis) - alebo tiež Karhunen-Loéve transformácie opísané v (Bishop, 2006), je štandard techniky pre štatistické (štruktúrne) rozpoznávanie a spracovanie signálu pre redukciiu dát a extrakciu črt. Vzor v obraze väčšinou obsahuje nadbytočné informácie. Namapovaním na vektor črt je možné sa zbaviť redundancie, a napriek tomu zachovať väčšinu dôležitého informačného obsahu vzoru. Tieto extrahované príznaky majú veľkú úlohu pri rozlíšení vstupných vzorov.

Obraz tváre v 2D s rozmermi $X \times Y$ môže byť tiež považovaný za jednorozmerný vektor o rozmere XY . Napríklad obraz tváre z databázy Olivettin Research Labs s rozmermi 112×92 môže byť považovaný za vektor o rozmere 10 304 alebo za bod v 10

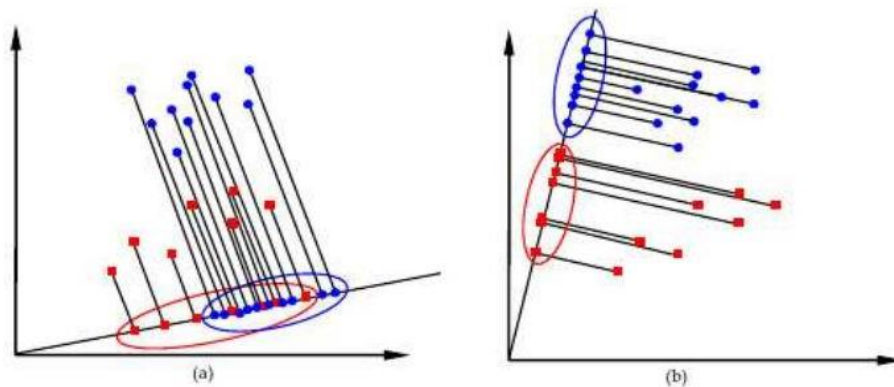
304 rozmernom priestore. Obrazy tváří, ktoré sú si celkovo podobné, nebudú náhodne distribuované v tomto obrovskom priestore, a preto môžu byť označené v relatívne rozmerovo malom pod-priestore. Hlavnou myšlienkou analýzy hlavných komponentov je nájdenie vektorov, ktoré najlepšie predstavujú rozdelenie obrazov tváří vo vnútri celého obrazového priestoru. Tieto vektory definujú pod-priestor obrazov tváří, ktorý nazývame priestor tváří. Každý z týchto vektorov má dĺžku XY , popisujúci obraz s rozmermi $X \times Y$, a je lineárnou kombináciou pôvodného obrazu tváre.

Lineárna diskriminačná analýza (Linear Discriminant Analysis) - popísaná v (Bishop, 2006) prekonáva obmedzenia metód založených na vlastných vektoroch za použitia Fisherovho lineárneho diskriminačného kritéria. Toto kritérium sa snaží maximalizovať pomer medzi determinantom medzitriednej matice rozptylu premietnutých vzoriek a determinantom vnútornej matice rozptylu premietnutých vzoriek.

Fisherfaces zoskupuje do skupín obrázky z rovnakej triedy a oddeľuje obrázky z rôznych tried. Obrázky sú premietané z M^2 rozmerného priestoru do D rozmerného priestoru, kde D je počet tried obrázkov. Napríklad uvažujme dve sady bodov vo dvojrozmernom priestore, ktoré sú premietané do jednej línie. V závislosti na smere línie môžu byť body buď zmiešané (Obrázok 9a) alebo oddelené (Obrázok 9b). Fisherov diskriminant nájde líniu, ktorá najlepšie rozdeľuje body. Pre rozpoznanie vstupného testovacieho obrazu je premietaný testovací obraz porovnaný s každým premietnutým obrazom na trénovanie a testovací obraz je určený ako najbližší obraz na trénovanie.

Rovnako ako vlastný priestor projekcie sú obrazy na trénovanie premietnuté do pod-priestoru. Testovacie obrazy sú premietnuté do rovnakého pod-priestoru a sú určené za použitia porovnávacích meradiel. Rozdiel je v počítaní pod-priestoru. Na rozdiel od metódy PCA, ktorá extrahuje príznaky najlepšej reprezentácie obrazu tváre, skúša metóda LDA nájsť pod-priestor, ktorý najlepšie popisuje rozdielne triedy tváří (Obrázok 9). V rámci vnútornej rozptylovej matice, tiež nazývanej intra-personal, sú reprezentované rozdiely v podobe rovnakého jedinca v dôsledku rozdielneho osvetlenia a výrazu tváre, zatiaľ čo v medzitriednej rozptylovej matici, nazývanej tiež extra-personal, sú reprezentované rozdiely v podobe odlišnej totožnosti. Použitím tejto metódy nájdeme smery projekcie, ktoré na jednej strane maximalizujú vzdialenosť medzi obrazmi tváří rôznych tried a na strane druhej minimalizujú vzdialenosť medzi obrazmi tváří rovnakej

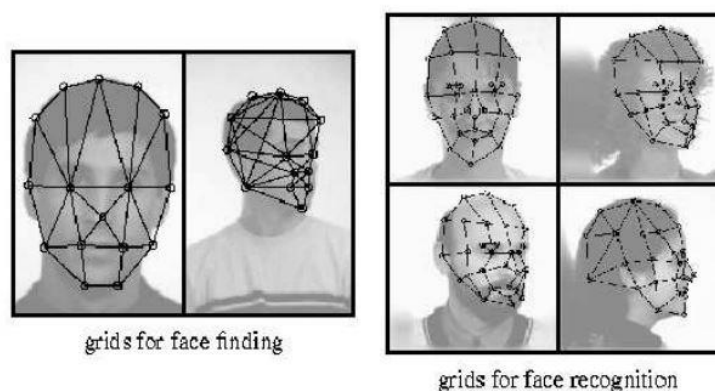
triedy. Inými slovami sa jedná o maximalizáciu medzitriednej rozptylovej matice Zb a zároveň minimalizáciu vnútornej rozptýlenej matice Sw v premietanom podpriestore.



Obrázok 9 a) zmiešané body pri premietnutí na priamku b) oddelené body pri premietnutí na inú priamku (Delac & Grgic, Face Recognition., 2007)

3.2.2 Elastické zhukové grafy

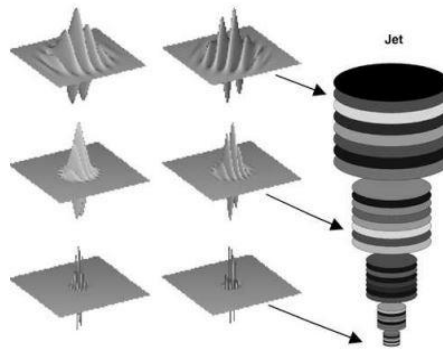
Všetky ľudské tváre zdieľajú podobnú topologickú štruktúru. V elastických zhukových grafoch (Elastic Bunch Graph Matching) (Wiskott, Fellous, & Krüger, 1997) sú tváre reprezentované ako grafy, kde sú uzly umiestnené vo východiskových bodoch. Hrany nesú informáciu o vzdialenosti a uzly sú označené ako wavelet koeficienty daného pixelu. Tento model grafu možno potom použiť pre generovanie obrazového grafu. Model môže byť posunutý, otočený, deformovaný pri porovnávaní alebo sa môže meniť mierka. Vďaka tomu táto metóda funguje, aj v prípade, že sú veľké rozdiely v obraze. Napríklad vplyvom osvetlenia či iného výrazu tváre.



Obrázok 10 Elastický porovnávací diagram aplikovaný na obrázok tváre (Wiskott, Fellous, & Krüger, 1997)

Obrázok 11 predstavuje zobrazené príklady Gáborovej vlnkovej transformácie o rôznych veľkostiach a priestorovej orientácie, ktorá sa používa na analýzu digitálnych

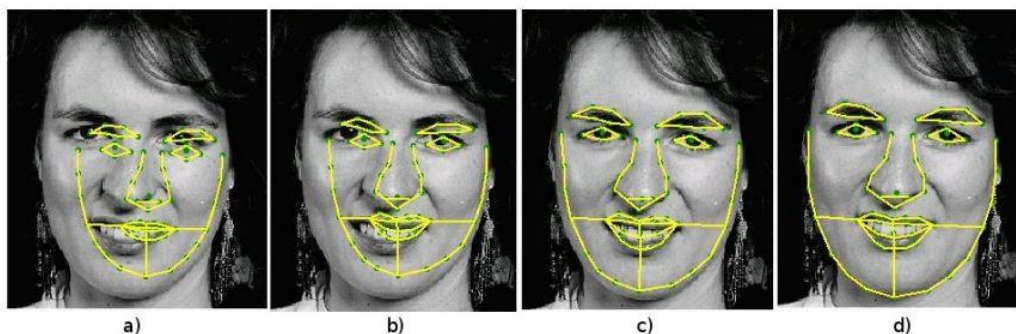
fotografií. Pre všetky body sú výsledky veľkosti a orientácia usporiadané do vektora rysov zvaného „jet“. Tento špeciálny typ metódy je možné použiť v rámci ktorejkoľvek fázy – či už je to detekcia, extrakcia alebo klasifikácia. Dôvodom je, že z Gáborovej vlnky sa v podobe Gaborovho filtra najčastejšie používajú ako detektor hrán. (Magdin M. , 2016)



Obrázok 11 Príklad Gáborovej vlnkovej transformácie (Loos, Wicczorek, & Würtz1, dátum neznámy)

3.2.3 Aktívny model vzhľadu (AAM)

Aktívny model vzhľadu (Active Appearance Model) (Cootes, 1998) je štatistický model, ktorý spája model získaný kombináciou tvarových vlastností s modelom vzhľadových zmien normalizovaného tvaru tváre. AAM obsahuje štatistický model, ak existuje zobrazenie tvaru a hodnôt intenzity objektu záujmu, ktorý sa môže zovšeobecniť na takmer akýkoľvek platný vzor. Zodpovedajúci obraz vyžaduje nájdenie parametrov modelu s najmenšou odchýlkou medzi obrazom a modelovým vzorom zobrazeným do obrazu (Obrázok 12).



Obrázok 12 Aktívny model vzhľadu a) inicializácia b) pozícia po prvej iterácii c) pozícia po druhej iterácii d) konvergované (Cootes, 1998)

3.3 Metódy klasifikácie emócií

Klasifikácia predstavuje poslednú fázu automatickej analýzy emócií, pričom jej úlohou je správne jednotlivé emočné výrazy vyhodnotiť napr. podľa Ekmanovej kategorizácie na základe extrahovaných príznakov. Nie je jednoduché povedať, ktorá metóda klasifikácie má univerzálne použitie. Všetko závisí od predchádzajúcej fázy - spôsobu extrakcie jednotlivých príznakov. Často sa preto metódy klasifikácie medzi sebou aj kombinujú, prípadne sa používajú už v prvej fáze - detekcii. V nasledujúcich podkapitolách uvádzame popis jednotlivých metód klasifikácie, ktoré sa presadili najviac. (Magdin M. , 2016)

3.3.1 Skrytý Markov Model

Skrytý Markov model (z angličtiny „Hidden Markov Model“ alebo HMM) je štatistický Markov model, ktorý modeluje systém za predpokladu, že ide o Markov proces so skrytými (nepozorovanými) stavmi. HMM môže byť najjednoduchšie znázornený pomocou dynamickej Bayesovej siete. (Magdin M. , 2016) Matematické základy modelu vyvinul L. E. Baum spolu so svojím tímom spolupracovníkov (Baum, 1966). Problematika veľmi úzko súvisí s predchádzajúcou prácou Ruslana L. Stratonoviča, ktorý pracoval na lineárnom probléme filtrovania (Stratonovich, 1960) a ako prvý popísal dopredne-spätný algoritmus.

V jednoduchších Markových modeloch (ako je Markov reťazec), je stav systému viditeľný pozorovateľovi, teda pravdepodobnosť zmeny stavu je jediný parameter modelu. Naopak v skrytých Markových modeloch stav nie je pozorovateľovi viditeľný, ale výstup, ktorý je na stave závislý, viditeľný je. Každý stav má pravdepodobnostný vplyv na výstup systému. Teda postupnosť výstupov skrytého Markovovho modelu vypovedá o postupnosti vnútorných stavov, ktorá túto postupnosť vygenerovala. Prívlastok skrytý sa teda vzťahuje na postupnosť vnútorných stavov, ktorými model prešiel, nie na parametre modelu (model sa nazýva skrytý, hoci sú jeho parametre dané presne a sú známe).

Skryté Markove modely sú známe najmä na poli rozpoznávanie časových vzorov. Patrí medzi nich napríklad rozpoznávanie reči, rúk písaného písma, gest (Starner & Pentland, 1995) a emócií. Využitie nachádza tiež v bio informatike.

Skryté Markove modely možno považovať za zovšeobecnenie zmiešaných modelov, v ktorých nie sú skryté premenné (alebo latentné premenné) nezávislé jedna na druhej, ale naopak sú navzájom spojené Markovovým procesom. V poslednej dobe boli skryté

Markove modely zovšeobecnené na párové Markove modely a tripletové Markove modely, ktoré umožňujú využitie modelu aj na komplexnejšie dátové štruktúry (Pieczyński, 2007) a modelovanie nestacionárnych dát (Boudaren, Monfrini, & Pieczyński, 2012).

HMM využíva pravdepodobnosť prechodu z jedného stavu do druhého v určitom časovom okamihu. Preto využitie HMM má veľký význam vtedy, ak uvažujeme o klasifikácii emócií subjektu, ktorého výraz je snímaný prostredníctvom videokamery a teda môže byť vyhodnocovaný real-time (časová postupnosť je reprezentovaná tvárovými pohybmi subjektu a teda je prirodzené modelovať napríklad každý výraz očí pomocou HMM) (Magdin M. , 2016).

3.3.2 Bayesov klasifikátor

S problematikou HMM, ktorá bola rozpracovaná v predchádzajúcej kapitole súvisí aj pojem Bayesova sieť. Problematika Bayesovej siete bola skúmaná od roku 1960, pôvodne jej využitie bolo smerované na oblasť spracovania textu - optické rozpoznávanie znakov a klasifikácia dokumentov.

HMM často býva charakterizovaný práve prostredníctvom jednoduchšej Bayesovej siete, ktorá pracuje na princípoch Bayesovho klasifikátora. Naivný Bayesov klasifikátor sa aj napriek svojej jednoduchosti používa dodnes, pretože svojou rýchlosťou výpočtu často prekonáva sofistikovanejšie metódy klasifikácie (napríklad SVM). Možnosťami využitia Bayesovho klasifikátora pri klasifikovaní emócií sa zaoberali viacerí autori odborných publikácií. Najvýznamnejšie výsledky však dosiahol Cohen (pre prehľad pozri Cohen et al., 2003) a Valenti (pre prehľad pozri Azcarate et al., 2005). Cohen použil Bayesov klasifikátor v prípade detekcie tváre v obraze, Valenti však jeho myšlienku rozvíjal ďalej. Z pôvodného výskumného tímu Azcarate et al., sa ďalej myšlienkou vytvorenia automatického systému pre rozpoznávanie tváre a jej častí zaoberal Roberto Valenti. Spolupracoval na rozvoji tohto systému s profesorom Theo Gevers, pričom ako prví navrhli aplikovanie Bayesovho klasifikátora na vyhodnocovanie emócií. Ich zrealizovaný systém ako jediný spĺňa podmienky ideálneho systému - nezávisí od svetelných podmienok, ale zase na druhej strane subjekt nemôže byť vzdialený od webkamery viac ako 1 m. Od roku sa tento projekt snaží profesor Valenti implementovať do platformy SecondLife - musíme poznamenať, že úspešne (pre prehľad pozri <http://sightcorp.com/demos/>) (Magdin M. , 2016).

3.4 OpenCV

OpenCV je otvorený projekt počítačového videnia (Open-source Computer Vision), ktorý je k dispozícii ako knižnica funkcií pre spracovanie obrazového vstupu v reálnom čase.

Počítačové videnie (Computer Vision) je názov odvetvia výpočtovej techniky a vývoja softvéru, ktorý sa zaoberá vývojom zariadení schopných získať informáciu zo zachyteného obrazu. Najčastejšie ide o rozpoznávanie tvárí alebo registračných značiek automobilov, ale všeobecne ide o celú problematiku spracovanie videa, obrazu z viacerých zdrojov alebo napríklad obrazovej analýzy materiálov.

Odbor počítačového videnia je v poslednej dobe na vzostupe, k čomu dopomáhajú ako lacnejšie a stále dokonalejšie kamerové systémy, tak aj ľahká dostupnosť potrebného výpočtového výkonu. Zároveň je nutné spomenúť tiež čoraz ľahšiu dostupnosť rôznych knižníc zameriavajúcich sa práve na spracovanie obrazu. Medzi posledné menované činitele patrí aj OpenCV, ktorý tiež stojí za rastom aplikácií s počítačovým videním a vývojárom na celom svete umožňuje ľahký vstup do problematiky celého systému.

So svojím zameraním na spracovanie v reálnom čase, pomáha OpenCV študentom i profesionálom jednoducho realizovať svoje pôvodne ťažko uskutočniteľné projekty. Vďaka tomu, že v jedinej, navyše voľne dostupnej knižnici sú k dispozícii funkcia počítačového videnia aj strojového rozpoznávania predmetov, je dnes táto vyspelá technológia, ktorá bola ešte donedávna výsadou iba v niekoľkých vyspelých výskumných centrách, dostupná naozaj každému záujemcovi.

Knižnica OpenCV obsahuje viac ako 500 optimalizovaných algoritmov a pod BCD licenciou je dostupná zadarmo aj pre komerčné využitie na systémoch Linux, Mac a Windows. Vďaka tomu sú podľa dostupných informácií v súčasnosti knižnice OpenCV využívané prakticky po celom svete a webové stránky projektu zaznamenali už viac ako 3 milióny sťahnutí.

Medzi aplikácie, ktoré je možné s OpenCV realizovať, je možné zaradiť napríklad systémy inteligentnej spolupráce medzi človekom a počítačom (z angličtiny „Human Computer Interface“ - HCI), identifikáciu objektov, segmentáciu a rozpoznávanie predmetov, rozpoznávanie tvárí a gest, sledovanie pohybu alebo pohybové ovládanie a samozrejme ďalšie aplikácie mobilnej robotiky.

3.5 Viola-Jones detektor

Objektový detektor Viola-Jones prvýkrát predstavili P. Viola a M. Jones v roku 2001 (Viola & Jones, Robust Real-time Object Detection, 2001). Jedná sa o detektor objektov pracujúci s obrazmi v odtieňoch šedej farby, ktorý sa skladá z troch základných častí: integrálneho obrazu, Haar kaskády a klasifikačného algoritmu AdaBoost. Výhodou tohto detektora je rýchlosť, dostatočná spoľahlivosť ako i značná nezávislosť na osvetlení a veľkosti sledovaného objektu. Z týchto dôvodov je v praxi tento detektor často používaný napríklad pri detekcii tváří a zároveň vzniká veľké množstvo jeho modifikácií (napr. využitie iné ako Haar kaskády, nahradenie algoritmu AdaBoost algoritmom GentleBoost a iné) (Friedman, Hastie, & Tibshirany, 2000).

AdaBoost (názov je skratkou pre Adaptive Boosting) je klasifikačný algoritmus (Šochman, 2005), ktorý vychádza z metódy strojového učenia zvaného boosting. Cieľom metódy boosting je zlepšenie klasifikačnej presnosti ľubovoľného algoritmu strojového učenia. Základom je vytvorenie viacerých klasifikátorov označovaných ako slabí žiaci (z angličtiny - weak learners). Tieto klasifikátory vznikajú pomocou výberu vzoriek zo základnej trénovanej množiny. Prvý klasifikátor má presnosť len o málo väčšiu ako je presnosť odhadu (t.j. viac ako 50% v prípade dvojstavového klasifikátora). Postupne sú potom pridávané ďalšie klasifikátory s obdobnou mierou presnosti, čím je vygenerovaný súbor klasifikátorov označovaný ako silný žiak (z angličtiny - strong learner), ktorého celková klasifikačná presnosť je ľubovoľne vysoká vzhľadom k vzorkám v trénovanej množine - klasifikácia bola zosilnená (z angličtiny - boosted) (Ethem, 2004).

Klasifikačný algoritmus AdaBoost teda pre učenie využíva slabé klasifikátory $h_i(x)$, ktoré sú vybrané z množiny klasifikátorov H , a ktorých lineárnou kombináciou vzniká nelineárny silný klasifikátor $H(x)$. Vstupom algoritmu je trénovacia množina S , ktorá je zložená z dvojíc (x_i, y_i) , kde x_i je získaná hodnota príznaku a y_i je trieda zodpovedajúca príznaku i , $y_i \in \{-1, 1\}$ pre $i = 1, \dots, M$, kde M je veľkosť trénovacej množiny. AdaBoost na rozdiel od základného algoritmu „boostingu“ používa váženie trénovacej množiny váhami D_t , ktoré sú zo začiatku nastavené rovnomerne a v každom cykle algoritmu sa potom vykonávajú nasledovné kroky:

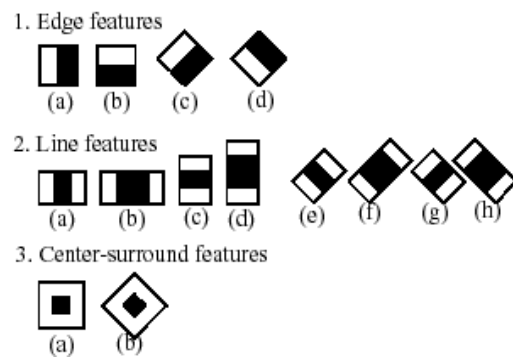
- Výber slabého klasifikátora s najmenšou chybou klasifikácie pri daných váhach D_t ,
- overenie, že chyba klasifikátora neprekročila hodnotu 0,5,
- výpočet koeficientu slabého klasifikátora v lineárnej kombinácii $H(x)$,

- aktualizácia jednotlivých váh D_t trénovacej množiny.

Ako už bolo uvedené, slabý klasifikátor je vyberaný tak, aby jeho presnosť klasifikácie bola o niečo lepšia, ako keby vykonával klasifikáciu náhodne. Ak teda chyba ε_t prekročí hodnotu 0,5, potom táto požiadavka nie je splnená a nie je teda zaručené, že algoritmus bude konvergovať. Aktualizácia váh zapríčini, že váha zle klasifikovaných meraní sa zväčší a váha dobre klasifikovaných sa zmenší. To znamená, že v nasledujúcom kroku bude hľadaný slabý klasifikátor, ktorý bude lepšie klasifikovať doposiaľ chybné vykonané merania (Viola & Jones, 2001). Adaboost teda redukuje trénovanie chyby exponenciálne v závislosti na rastúcom počte klasifikátorov. Zvyšovanie počtu klasifikátorov v skupine ale môže viesť k tzv. „pretrénovaniu“ (t.j. strate schopnosti generalizovať vplyvom prílišného zamerania klasifikátorov na rozoznávanie iba konkrétnych trénovacích dát). Simulačné experimenty však ukázali, že k tomu dochádza relatívne zriedka aj pre extrémne vysoké hodnoty počtu klasifikátorov (Rosenblatt, 1958).

Haar kaskády - Ako bolo spomenuté v predchádzajúcej časti, vstupom procesu učenia klasifikačného algoritmu je množina príznakov S . Čím je táto množina príznakov obsiahlejšia, tým existuje väčšia pravdepodobnosť výberu slabého klasifikátora s vyššou mierou presnosti (samozrejme s ohľadom na vzájomnú koreláciu jednotlivých príznakov). Snahou detektora Viola-Jones je získať veľký rad jednoduchých príznakov s minimálnymi výpočtovými nárokmi. Takýmto typom príznakov sú príznaky založené na princípe, ktorý je podobný definícii Haar kaskád (tzv. Haar-like features), viď Obrázok 13. Hodnota takéhoto príznaku sa tak vypočíta ako súčet pixelov obrazu zodpovedajúcej svetlej časti, od ktorej je odpočítaná suma pixelov zodpovedajúcej tmavej časti. Tieto vlnky môžu byť tvorené dvoma (hranový príznak), tromi (čiarový príznak) či štyrmi (diagonálny príznak) obdĺžnikovými oblasťami. Jednotlivé príznaky sú použité na celý vstupný obraz, pričom zároveň dochádza k zmene veľkostí jednotlivých príznakov (t.j. Veľkosti jednotlivých obdĺžnikov) z veľkosti 1×1 až na veľkosť zodpovedajúcu vstupnému obrazu. To znamená, že pre vstupný obraz s rozmermi 19×19 dostávame približne 64 tisíc hodnôt príznakov, ktoré sú vstupom pre učiaci proces klasifikačného algoritmu AdaBoost. Ten z nich potom vyberie len určité malé množstvo príznakov spoločne s rovnakým počtom natrénovaných slabých klasifikátorov, pomocou ktorých možno vstupný obraz vhodne klasifikovať (rovnaký príznak sa vo výslednom silnom klasifikátore môže vyskytovať aj niekoľkokrát, avšak zakaždým s iným nastavením slabého klasifikátora). Iba toto malé množstvo príznakov je potom použité pri samotnej detekcii. Uvedené typy príznakov patria

k tzv. Základným príznakom, v súčasnosti sa používajú aj ďalšie typy zo základných príznakov odvodené od autorov (Lienhart & Maydt, 2002).



Obrázok 13 Príznačky podobné Haar kaskáde (OpenCV, 2017)

4 NÁVRH A RIEŠENIE Č.1

V nasledovnej kapitole popíšeme postup pri vývoji nášho riešenia systému pre rozpoznávanie a vyhodnocovanie ľudských emócií v reálnom čase na základe výrazu tváre, získaného za pomoci webovej kamery. Systém bol vyvíjaný na notebooku so 64 bitovým operačným systémom Windows 10 Pro, obsahujúcim procesor Intel(R) Core(TM) i5-4200M CPU @ 2,50 GHz 2,50 GHz a 6 GB operačnej pamäte. Notebook disponuje webkamerou s rozlíšením 1280x720 pixelov a bude použitá pre účely testovania riešení.

Riešenie bude pozostávať z troch fáz:

1. detekcie tváre,
2. extrakcie črt,
3. klasifikácie emócií.

4.1 Detekcia tváre v jazyku Java

V prvej fáze riešenia sme sa zamerali na detekciu tváre v obraze. Po zhodnotení teoretických východísk práce, ktoré sú popísané v kapitole **Error! Reference source not found.**, sme sa rozhodli na detekciu tváre použiť Haar kaskády a ako implementačný jazyk sme si zvolili Javu. Tento programovací jazyk má v tomto prípade výhodu hlavne v tom, že by výslednú aplikáciu nebolo potrebné kompilovať pre rôzne operačné systémy zvlášť. Rozhodli sme sa využiť otvorenú knižnicu OpenCV, verziu 3.2 ktorá obsahuje implementáciu Viola-Jones detektora ako aj Haar kaskády.

Ako vývojové prostredie bol zvolený Eclipse Neon.2 vzhľadom k predchádzajúcim skúsenostiam s daným vývojovým prostredím.

Pred zahájením programovacej fázy bolo potrebné správne nastaviť „Java Build Path“ – do knižníc bolo potrebné pridať „JRE System library“ (Java Runtime Environment), v tomto prípade bola použitá verzia jre1.8.0_122. Ďalej bolo potrebné do zoznamu knižníc pridať umiestnenie stiahnutej knižnice OpenCV-3.2

Cieľom bolo dosiahnuť detekciu ľudskej tváre za pomoci webkamery ako zdroja dát v reálnom čase. Vzhľadom k použitiu metódy detekcie pomocou Haar kaskád bolo od začiatku zrejmé, že systém bude vykazovať najlepšie výsledky pri detekcii tváre, ktorá je natočená k webkamere priamo a nie z boku. Predpokladali sme, že pri horizontálnom natočení tváre môže mať systém problémy správne detegovať tvár.

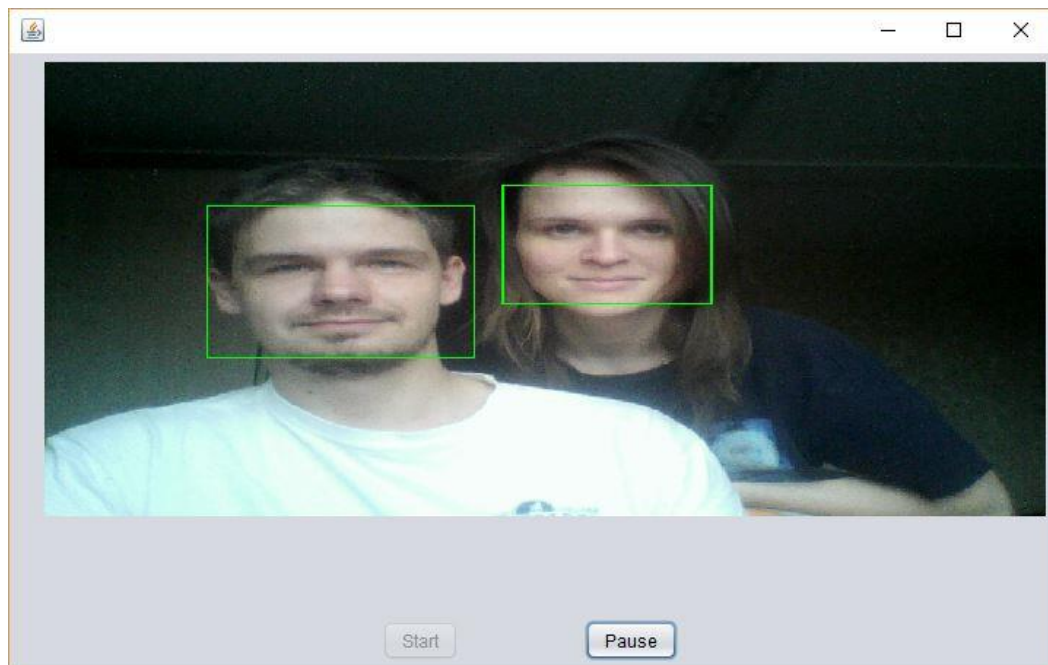
4.2 Používateľské rozhranie

V tejto fáze riešenia problematiky detekcie tváre sme sa zamerali na čo najväčšiu jednoduchosť používateľského prostredia.

Rozloženie sme navrhli nasledovne (Obrázok 14): celý obsah aplikácie bude zobrazený v jednom okne. Najväčšiu časť plochy okna bude zaberat' aktuálny výstup z webkamery. Ovládacie prvky boli navrhnuté nasledovne:

- Tlačidlo „Start“ pre zapnutie zobrazenia výstupu z webkamery a zároveň spustenie detekcie tváre
- Tlačidlo „Pause“ pre pozastavenie chodu programu.

Ako predvolené nastavenie po spustení programu bude zobrazenie okna a program bude čakať na užívateľa, aby spustil detekciu.



Obrázok 14 Používateľské rozhranie aplikácie na detekciu tváre

4.3 Realizácia samotného systému

Po nastavení vývojového prostredia a nevyhnutnom konfigurovaní projektu v Eclipse sme pristúpili k samotnej realizácii. V nasledujúcej ukážke sú uvedené konkrétne knižnice a sekcie knižníc, s ktorými v našom systéme pracujeme.

```
import java.awt.Graphics;  
import java.awt.Image;  
import java.awt.image.BufferedImage;  
import java.io.ByteArrayInputStream;  
import javax.imageio.ImageIO;
```

```

import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfByte;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;
import org.opencv.videoio.VideoCapture;

```

Ukážka 1 Knižnice použité na implementáciu detekcie tváre

Následne sme vytvorili potrebné objekty pre detekciu tváre. Objekt `webSource` má na starosti získanie jednotlivých snímok z webkamery. Tie sú dostupné pomocou objektu `frame`. V ukážke vidíme spôsob priradenia dôležitého zdroja - `haarcascade_frontalface_alt.xml` v ktorom sa nachádza Haar kaskáda o rozmere 20x20 pixelov triede `CascadeClassifier` so svojou inštanciou `faceDetector`. A napokon objekt `faceDetections` nesie informáciu o pozícií detegovanej tvári.

```

VideoCapture webSource = null;
Mat frame = new Mat();
MatOfByte mem = new MatOfByte();
CascadeClassifier faceDetector = new CascadeClassifier(
FaceDetection.class.getResource("haarcascade_frontalface_alt.xml").getPath().substring(1));
MatOfRect faceDetections = new MatOfRect();

```

Ukážka 2 Dôležité objekty systému pre detekciu tváre

Za pomoci nasledovnej ukážky môžeme demonštrovať samotnú detekciu tváre. Kľúčová je metóda `detectMultiScale()`, ktorá patrí triede `CascadeClassifier` a ktorá deteguje objekty rôznych veľkostí zo vstupného snímku. Metóda vracia detegované objekty ako zoznam obdĺžnikov, ktoré je možné následne vykresliť v okne aplikácie.

```

if (webSource.grab()) {
    try {
        webSource.retrieve(frame);
        Graphics g = jPanell.getGraphics();
        faceDetector.detectMultiScale(frame, faceDetections);
        for (Rect rect : faceDetections.toArray()) {
            Imgproc.rectangle(frame, new Point(rect.x, rect.y),
                new Point(rect.x + rect.width, rect.y + rect.height),
                new Scalar(0, 255, 0));
        }
        Imgcodecs.imencode(".bmp", frame, mem);
        Image im = ImageIO.read(new
        ByteArrayInputStream(mem.toArray()));
        BufferedImage buff = (BufferedImage) im;
        if (g.drawImage(buff, 0, 0, getWidth(), getHeight() - 150, 0,
        0, buff.getWidth(), buff.getHeight(), null)) {
            if (runnable == false) {
                System.out.println("Paused ..... ");
            }
        }
    }
}

```



```

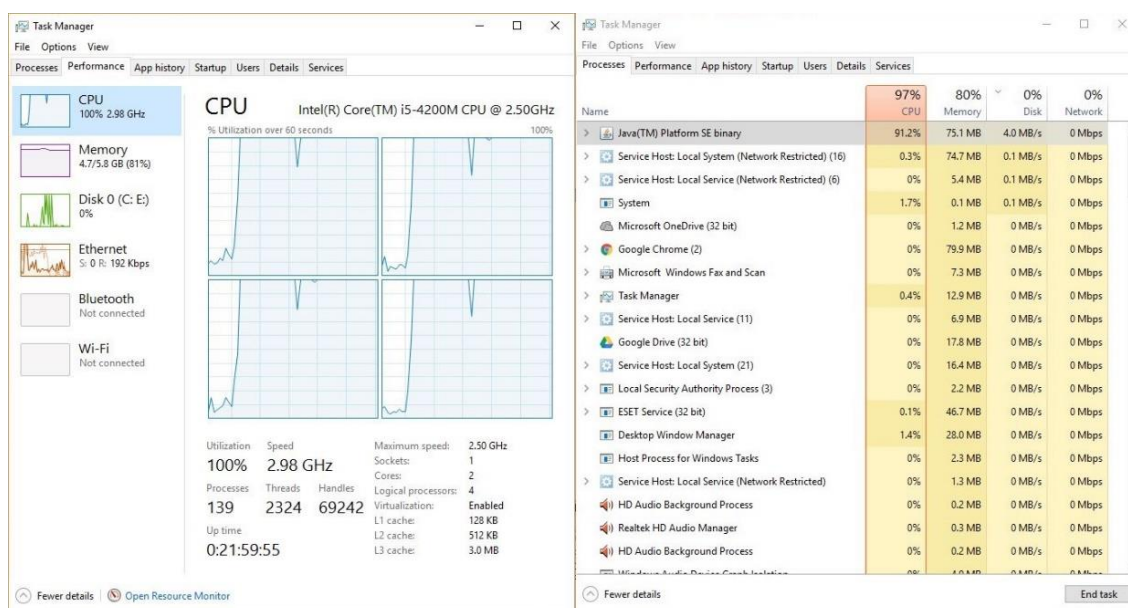
        this.wait();
    }
}
} catch (Exception ex) {
    System.out.println("Error");
}
}

```

Ukážka 3 Proces detekcie tváre a vykresľovanie obdĺžnikov pre nájdené tváre

4.4 Zhrnutie

Systém pre detekciu tváre sa nám podarilo úspešne implementovať. Tváre boli detegované s 98% úspešnosťou, teda s veľmi veľkou presnosťou. Problém detekcie nastával až pri natočení hlavy o viac ako 15 stupňov. Systém je schopný za bežných svetelných podmienok detegovať tvár až na 7 metrov. Minimálnou vzdialenosťou detekcie za predpokladu, že žiadna časť tváre nebola zakrytá a v zábere bola celá tvár, je 15 cm. V ojedinelých prípadoch systém detegoval falošnú tvár, väčšinou v tmavšej časti snímky, z čoho môžeme usúdiť, že za falošnú detekciu môže šum v obraze. Najväčším úskalím tohto riešenia však bola výpočtová náročnosť. Zariadenie uvádzané na začiatku kapitoly NÁVRH A RIEŠENIE č.1 obsahuje procesor s dvomi fyzickými jadrami, softvérovo rozdelenými na štyri jadrá, a pri spustenej detekcii tvare bol procesor vytiažený na plný výkon. Samotná aplikácia si na svoj chod vyžadovala v priemere 93.8% výkonu procesora pri detekcii jednej tváre (Obrázok 15). V prípade, že detektor v danom momente nedetegoval žiadnu tvár, chod aplikácie si vyžadoval priemerne 89.3%. Na základe vysokých nárokov na výpočtový výkon sme sa rozhodli od tohto riešenia upustiť a hľadať výhodnejšie riešenie.



Obrázok 15 Vyťaženie systémových zdrojov počas detekcie tváre

5 RIEŠENIE Č.2, VÝSLEDKY A ICH ZHODNOTENIE

V nasledujúcej kapitole sa podrobne pozrieme na finálne riešenie vyhodnocovania emocionálneho stavu používateľa pomocou web kamery. Chceme však poznamenať, že vývoj tohto softvéru nekončí odovzdaním diplomovej práce. Máme nápady na vylepšenia a funkcionality a z tohto dôvodu sa verzia programu odovzdaná spolu s prácou môže líšiť od verzie, ktorá bude prezentovaná na obhajobe diplomovej práce.

Z poznatkov z predošlej kapitoly sme sa poučili a hľadali sme alternatívne riešenia. Jedným z návrhov bol systém, ktorý za pomoci špeciálneho hardvéru – termo kamery pripevnenej k hlave používateľa by snímал oči z blízkej vzdialenosti. Použitie termo kamery by malo význam v rámci eliminácie detegovania ľudskej tváre z fotografie a systém by tak snímал naozaj iba živých ľudí. Systém by sa mohol zamerať na zreničky používateľa, merať ich veľkosť a za použitia HMM (Hidden Markov Model) klasifikovať emocionálny stav používateľa. Od tohto riešenia sme nakoniec upustili kvôli značným nevýhodám, ktoré by takýto systém priniesol. Za najväčšiu nevýhodu sme považovali zabránenie voľnému výhľadu používateľa, čo by bolo nepraktické obzvlášť v prípade nasadenia systému v oblasti vzdelávania.

5.1 SDK od spoločnosti Affectiva

Affectiva je spoločnosť, ktorá pôsobí od roku 2009, vznikla na pôde univerzity MIT v oddelení Media Lab, a ich hlavným zameraním je „naučiť počítače rozumieť ľudským

emóciám“. Jej riešenia nachádzajú uplatnenie hlavne v reklamnom sektore. Počas svojho výskumu napríklad vystavovali ľudí video-reklamám a za pomoci webkamery snímali reakcie subjektov a analyzovali prežívané emócie. Na základe takéhoto pozorovania mohli potom vyhodnotiť reakcie na rovnakú reklamu v rôznych demografických zázemiach. Spoločnosť Affectiva na trh uviedla aj nositeľný bio-senzor, ktorý dokáže odhaliť emócie na základe vstupu z pokožky.

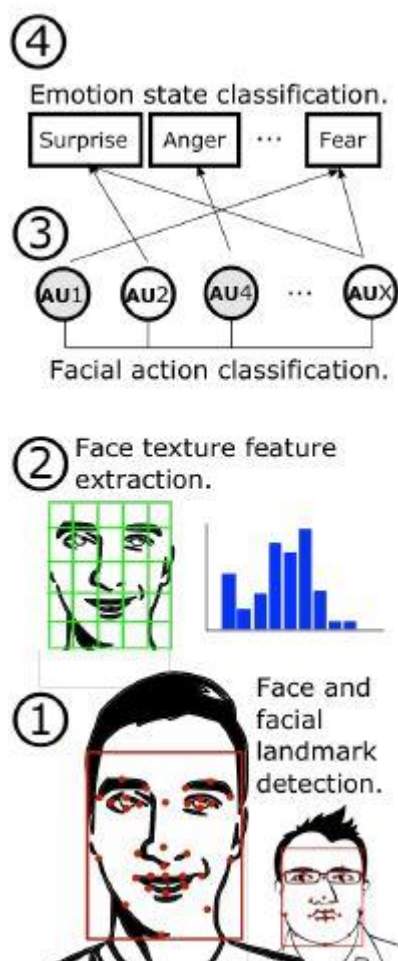
Spoločnosť prvý krát sprístupnila „Software Development Kit“ pre vývojárov v roku 2015. Aktuálna verzia SDK 3.4.1 pre operačný systém Windows bola vydaná v apríli 2017. Cieľom SDK je detegovať výrazy tváre a klasifikovať emócie s nimi spojené. SDK obsahuje komponenty, ktoré umožňujú integráciu s frameworkom .NET a Aplikáciami, ktoré boli napísané v jazyku C++. Na zabezpečenie správneho chodu SDK sú vyžadované nasledovné komponenty:

- **affdex-native**: hlavičkové súbory a knižnice C++ (verzie Release a Debug),
- **Addfex**: zostava menového priestoru .NET (verzie Release a Debug),
- **dáta**: klasifikačné dáta, ktoré pre svoj správny chod potrebujú Affdex knižnice ako aj affdex-native počas chodu programu.

Odporúčané hardvérové požiadavky sú nasledovné:

- procesor: 2GHz,
- pamäť RAM: 1 GB,
- voľné miesto na disku: 950 Mb.

SDK podporuje Windows 7 a vyššie. Pre správny chod vyžaduje Visual C++ Redistributable runtime for VisualStudio 2013 a Microsoft .NET framework v 4.0. SDK od spoločnosti Affectiva podporuje viacero platform, medzi ktoré patria okrem Windowsu aj Android (Java), iOS (Objective-C), Linux (C++), macOS (Objective-C), Unity a web (JavaScript) (Affectiva, 2017).



Obrázok 16 Reťazec automatizovaného spracovania tváre (McDuff, et al., 2016)

Na Obrázok 16 Obrázok 16 Reťazec automatizovaného spracovania tváre je možné vidieť rozšírený model spracovávaní výrazu tváre a to v nasledovných bodoch:

1. Detekcia tváre a detekcia významných črt tváre,
2. Extrakcia črt,
3. Klasifikácia akčných bodov tváre,
4. Modelovanie prototypov emócií s použitím EMFACS.

Detekcia tváre a detekcia významných črt tváre - vykonáva sa pomocou algoritmu detekcie tváre Viola-Jones (Viola & Jones, Rapid object detection using a boosted cascade of simple features., 2001). Detekcia orientačných bodov sa potom aplikuje na každý obdĺžnik ohraničujúci tvár na 34 identifikovaných orientačných bodov. Ak je konfidencia detekcie orientačných bodov pod prahovou hodnotou, ohraničenie sa ignoruje. Extrahované body tváre, pootočené hlavy a bod vyjadrujúci stred medzi očami pre každú tvár sú zobrazené v SDK.

Akcie tváre - Histogram Orientovaného Gradientu (z anglického „Histogram of Oriented Gradient“ - HOG) (Dalal & Triggs, 2005) sa extrahuje z oblasti záujmu obrazu definovaný orientačnými bodmi tváre. Algoritmus podporných vektorov (z anglického „Support Vector Machine“ - SVM), trénovaný na 10 000 manuálne kódovaných obrázkov tvári zozbieraných z celého sveta sa používa na vygenerovanie skóre od 0 do 100 pre každú akciu tváre. Podrobné informácie o systéme trénovania a testovania nájdete v (Senechal, D., & Kaliouby, 2015).

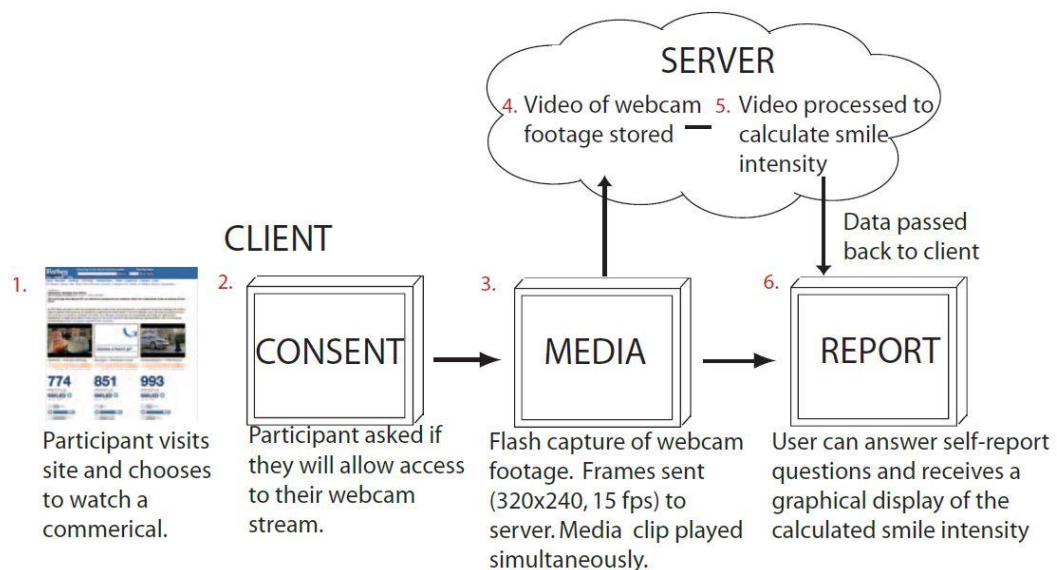
Emocionálne výrazy – v tomto prípade sú to hnev, strach, radosť, smútok, prekvapenie a pohrdanie sú založené na kombináciách akcií tváre. Toto kódovanie bolo postavené na emocionálnom systéme kódovania činnosti tváre EMFACS (Friesen & Ekman, 1983). Emocionálne výrazy dostávajú skóre od 0 (výraz chýba) do 100 (výraz je prítomný).

5.2 Databáza výrazov tváre od Affectiva

Počítačová klasifikácia výrazov tváre si vyžaduje rozsiahle množstvo dát, ktoré by mali odrážať rozmanitosť podmienok vyskytujúcich sa v skutočnom svete. Verejné databázy pomáhajú urýchliť pokrok výskumu tým, že poskytujú výskumným pracovníkom referenčný zdroj. Predstavujú komplexne označenú množinu platných spontánnych reakcií tváre zaznamenaných v prirodzenom prostredí cez internet. Za účelom zhromažďovania údajov sledovali diváci prostredníctvom internetu jednu z troch zámerne zábavných reklám na „Super Bowl“ a súčasne boli nahrávaní pomocou webovej kamery. Odpovedali na tri hodnotiace otázky o svojej skúsenosti. Určitá skupina divákov okrem toho poskytla súhlas na to, aby ich údaje boli verejne zdieľané s ostatnými výskumníkmi. Táto podskupina pozostáva z 242 video záznamov tvári (168 359 snímok) zaznamenaných v reálnych podmienkach. Databáza je kompletne označená nasledujúcimi údajmi:

1. Označenia po jednotlivých snímkach, ktoré vyjadrujú 10 symetrických FACS akčných jednotiek, 4 asymetrické (jednostranné) akčné jednotky FACS, 2 pohyby hlavy, úsmev, všeobecná expresivita, zlyhanie extrakcie črt tváre a pohlavie
2. umiestnenie 22 automaticky rozpoznaných orientačných bodov
3. vlastné odpovede na otázky: Bolo vám video povedomé? Máte chuť opäť sledovať stimulačné videá?
4. index základnej výkonnosti detekčných algoritmov na tejto databáze.

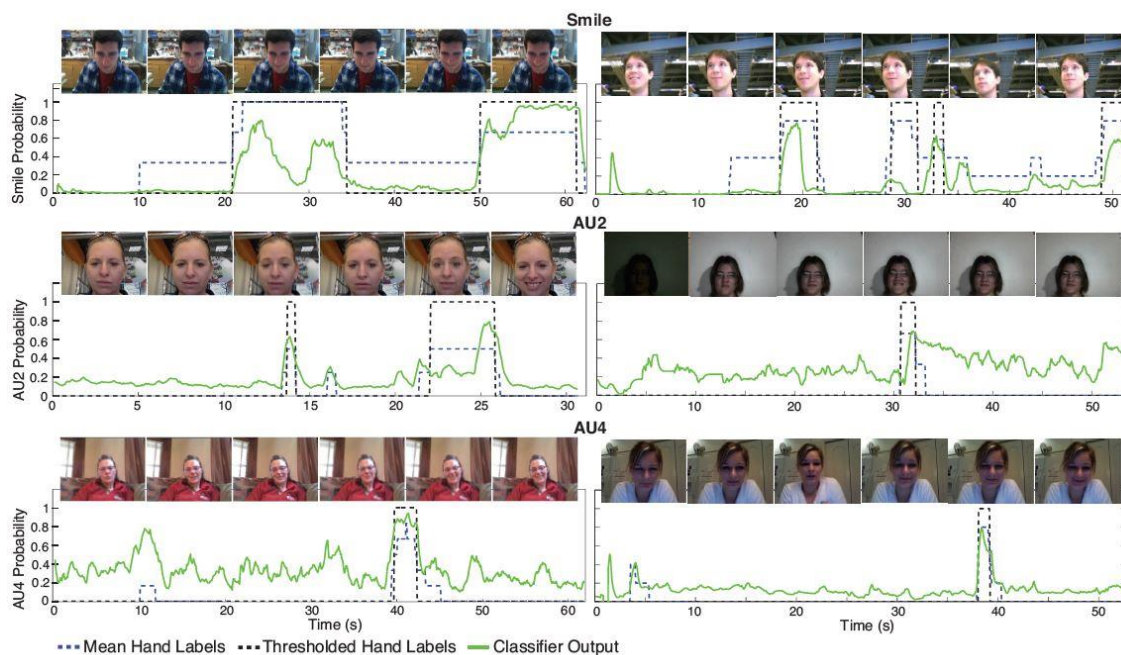
Tieto údaje sú k dispozícii na distribúciu výskumným pracovníkom prostredníctvom internetu. Licencia EULA sa nachádza na adrese: <http://www.affective.com/facial-expression-dataset-am-fed/>. (McDuff, a iní, 2013)



Obrázok 17 Prehľad princípu fungovania webového frameworku na zbieranie tvárových dát z verejnosti (McDuff, a iní, 2013)

Video z webkamery sa prenáša v reálnom čase na server, kde sa vykonáva automatická analýza výrazu tváre a výsledky sa vracajú naspäť do internetového prehliadača pre zobrazenie. Celé spracovanie videa prebehlo na strane servera. Zbieranie a vyhodnocovanie dát prebehlo v nasledujúcich krokoch:

1. účastníci navštívili webstránku a vybrali si reklamu, ktorú si chceli pozrieť
2. obsah: účastníkovi je položená otázka, či je ochotný povoliť prístup ku svojej webkamere
3. médiá: získanie snímok z webkamery. Snímky sú odoslané na server v rozlíšení 320x240, 15 snímok za sekundu. Zároveň sa účastníkovi zobrazuje mediálny klip
4. uloženie snímok získaných z webovej kamery
5. video je spracované – vypočíta sa intenzita úsmevu
6. dáta sú odoslané naspäť klientovi
7. účastník si môže zvoliť, či je ochotný odpovedať na seba-hodnotiace otázky a je mu zobrazené grafické znázornenie vypočítanej intenzity úsmevu.



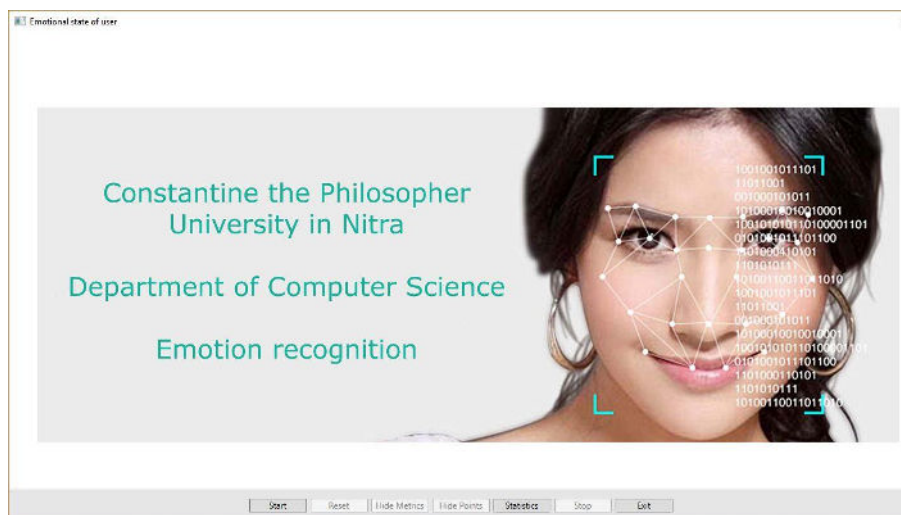
Obrázok 18 Príklad porovnania medzi predikciami klasifikátora (zelená línia) a ručne zadanými popismi (modrá a čierna prerušovaná línia) pre 6 videí vybraných z databázy (McDuff, a iní, 2013)

5.3 Používateľské rozhranie

V tejto fáze riešenia sme rozšírili pôvodné používateľské rozhranie o ďalšiu funkcionality. Všetky ovládacie prvky sú situované v hlavnom okne. Najväčšiu časť obsahu tvorí výstup z webkamery. Okrem tlačidiel „Start“ a „Stop“ boli pridané nasledovné prvky:

- Tlačidlo „Reset“ – slúži pre opätovnú inicializáciu snímania obrazu v prípade chyby
- Tlačidlo „Show/Hide Metric“ – umožňuje zobrazenie a vypnutie zobrazenia vyhodnocovania emocionálneho stavu používateľa v reálnom čase
- Tlačidlo „Show/Hide Points“ – slúži na prepínanie medzi zobrazením a vypnutím zobrazenia bodov, ktoré znázorňujú črty tváre
- Tlačidlo „Statistics“ – pomocou tohto prvku je možné nahliadnuť do štatistických údajov, ktoré program počas snímania zaznamenal
- Tlačidlo „Exit“ – slúži na vypnutie programu.

Program je nastavený tak, aby začal snímať hneď po spustení a úvodnej inicializácii. Grafické rozhranie bolo definované pomocou jazyka XAML. Nižšie je znázornené používateľské rozhranie aplikácie v pozastavenom stave.



Obrázok 19 Ukážka používateľského rozhrania aplikácie

5.4 Vývoj finálnej aplikácie

Finálna aplikácia je napísaná v jazyku C# a ako vývojové prostredie bolo zvolené Microsoft Visual Studio 2015. Pred začatím samotného programovania bolo potrebné nainštalovať SDK od Affectiva, ktoré je dostupné na stránke <https://download.affectiva.com/windows/AffdexSDK-3.4.1-1320-win32.exe> pre 32 bitový operačný systém alebo na stránke <https://download.affectiva.com/windows/AffdexSDK-3.4.1-575-win64.exe> pre 64 bitový operačný systém.

Po vytvorení vlastného „Sollution“ a projektu je potrebné pridať referenciu na knižnicu Affdex.dll, ktorá sa štandardne nachádza v priečinku „C:\Program Files\Affectiva\AffdexSDK\bin\release“. Je potrebné pridať aj knižnicu affdex-native.dll, ktorá sa nachádza v rovnakom priečinku.

5.4.1 Popis tried a metód aplikácie

Trieda FilePath.cs - V ďalšom kroku bola vytvorená trieda FilePath.cs obsahujúca jedinú metódu – GetClassifierDataFolder(), ktorá vracia cestu k databáze klasifikátorov.

```
class FilePath
{
    static public String GetClassifierDataFolder()
    {
        return "C:\\Program Files\\Affectiva\\AffdexSDK\\data";
    }
}
```

Ukážka 4 Trieda FilePath.cs

Trieda MainWindow.cs – Táto trieda sa stará o interakciu s používateľom, sú v nej implementované metódy na obsluhu akcií, ktoré sa majú vykonať po stlačení jednotlivých tlačidiel, ako aj inicializácia klasifikátorov, obsluhovanie výnimiek pri chybe, vykresľovanie obrazu získaného pomocou webkamery a ukladanie užívateľských nastavení. Popíšeme si niektoré udalosti, ktoré sa vykonávajú v metóde `Windows_Loaded()`. Z nastavení sa zapnú prednastavené klasifikátory výrazu tváre. Tie sú v našom prípade klasifikátory pre nasledovné emócie: šťastie, smútok, hnev, znechutenie, prekvapenie a strach. Zároveň sa určí a načíta posledné nastavenie pre zobrazovanie/skrytie zobrazovania tvárových črt a jednotlivých emócií, ktoré aplikácia detekuje.

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    EnabledClassifiers = Emotions.Settings.Default.Classifiers;
    canvas.MetricNames = EnabledClassifiers;
    ShowFacePoints = canvas.DrawPoints =
    Emotions.Settings.Default.ShowPoints;
    ShowMetrics = canvas.DrawMetrics =
    Emotions.Settings.Default.ShowMetrics;
}
```

Ukážka 5 Metóda Window_Loaded()

StartCameraProcessing() je veľmi dôležitá metóda, pretože sa v nej nastavuje zdroj obrazu, ktorý sa bude snímať a analyzovať. Po úvodnom nastavení, či je možné kliknúť na jednotlivé tlačidlá sa inicializuje „CameraDetector“ za pomoci niekoľkých konštantných celočíselných premenných. Konštanta `cameraId` označuje zariadenie, ktoré bude snímať obraz. Priradená hodnota 0 zodpovedá prednastavenej webovej kamere v zariadení, na ktorom aplikácia beží. Konštanta `numberOfFaces` udáva maximálny počet tvárí, ktoré bude aplikácia detegovať. Konštanta `cameraFPS` udáva počet snímok za sekundu, ktoré bude aplikácia získavať z webkamery. Konštanta `processFPS` označuje počet snímok za sekundu, ktoré budú spracované aplikáciou. Posledná konštanta pri inicializácii detektora je `Affdex.FaceDetectorMode`, ktorý môže nadobúdať hodnoty `SMALL_FACES` a `LARGE_FACES`. Hodnota by sa mala nastaviť podľa toho, či sa má aplikácia zamerať na tváre blízko, alebo ďalej od webkamery. Nasledujúce riadky kódu sa starajú o skrytie obrázku pozadia, zobrazenie obrazu získaného z webkamery, zobrazenie vrstvy, ktorá vykresľuje namerané hodnoty výrazov tváre a extrahované črty. Na záver sa spustia meranie celkového času detekcie. Zvyšok kódu sa stará o odchytenie chýb počas behu aplikácie a zobrazenie príslušných hlások.

```

private void StartCameraProcessing()
{
    try
    {
        btnStatistics.IsEnabled =
        btnStartCamera.IsEnabled = false;
        btnResetCamera.IsEnabled =
        Points.IsEnabled =
        Metrics.IsEnabled =
        btnStopCamera.IsEnabled =
        btnExit.IsEnabled = true;

        // Instantiate CameraDetector using default camera ID
        const int cameraId = 0;
        const int numberOfFaces = 10;
        const int cameraFPS = 30;
        const int processFPS = 30;
        Detector = new Affdex.CameraDetector(cameraId, cameraFPS,
        processFPS, numberOfFaces,
        Affdex.FaceDetectorMode.LARGE_FACES);

        //Set location of the classifier data files, needed by the
        SDK
        Detector.setClassifierPath(FilePath.GetClassifierDataFolder()
        );

        // Set the Classifiers that we are interested in tracking
        TurnOnClassifiers();

        Detector.setImageListener(this);
        Detector.setProcessStatusListener(this);

        Detector.start();

        // Show the camera feed and the data canvas
        background.Visibility = Visibility.Hidden;
        canvas.Visibility = Visibility.Visible;
        cameraDisplay.Visibility = Visibility.Visible;
        stopwatch.Start();
    }
    catch (Affdex.AffdexException ex)
    {
        if (!String.IsNullOrEmpty(ex.Message))
        {
            // If this is a camera failure, then reset the application to
            allow the user to turn on/enable camera
            if (ex.Message.Equals("Unable to open webcam."))
            {
                MessageBoxResult result = MessageBox.Show(ex.Message,
                "Error",
                MessageBoxButton.OK,
                MessageBoxImage.Error);
                StopCameraProcessing();
                return;
            }
        }
    }

    String message = String.IsNullOrEmpty(ex.Message) ? "Error
    encountered." : ex.Message;

```

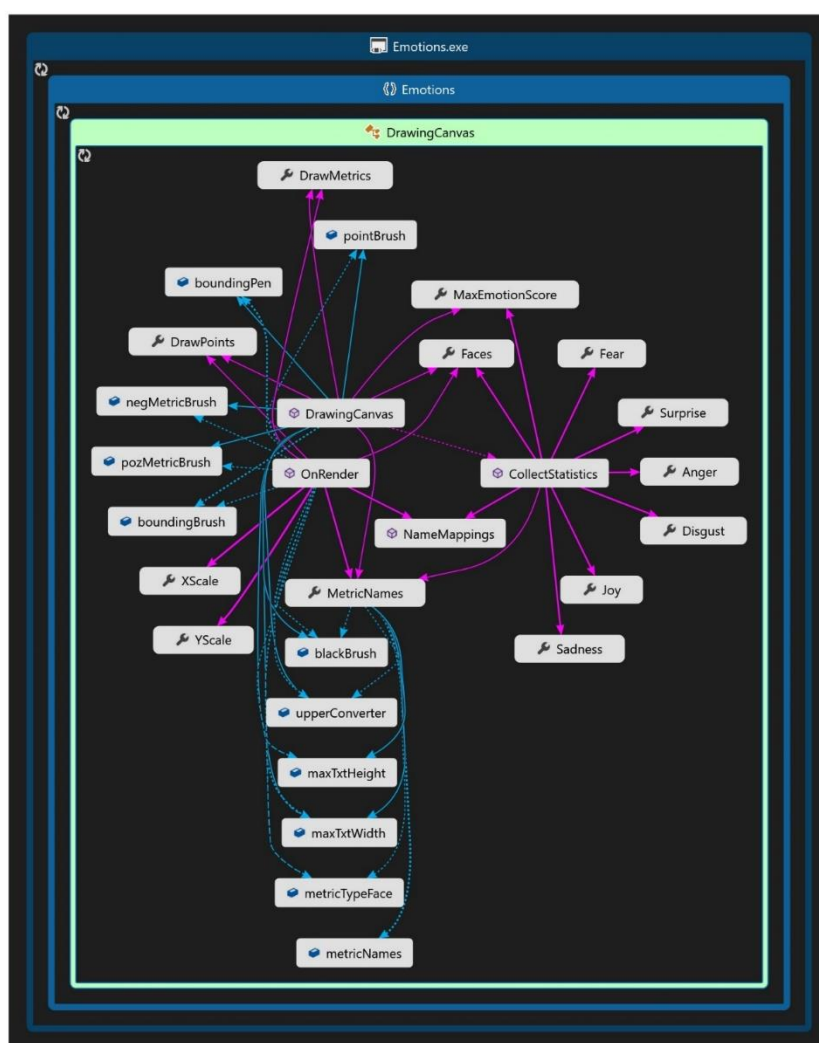
```

        ShowExceptionAndShutDown(message);
    }
    catch (Exception ex)
    {
        String message = String.IsNullOrEmpty(ex.Message) ? "Error
        encountered." : ex.Message;
        ShowExceptionAndShutDown(message);
    }
}

```

Ukážka 6 Metóda StartCameraProcessing()

Ako poslednú si popíšeme triedu **DrawingCanvas.cs**, ktorá má na starosti vykresľovanie nameraných hodnôt v reálnom čase, ako aj zber dát pre vyhodnotenie štatistík. Pre úplnosť uvádzame diagram triedy.



Obrázok 20 Diagram triedy DrawingCanvas.cs

CollectStatistics() je metóda, ktorá sa volá dva krát za sekundu a zbiera aktuálne dáta vyhodnotených emócií.

```

private void CollectStatistics(object source, ElapsedEventArgs e)
{

```

```

foreach (KeyValuePair<int, Affdex.Face> pair in Faces)
{
    Affdex.Face face = pair.Value;
    face = pair.Value;
    MaxEmotionScore = MaxEmotionScore + (100 * Faces.Count);
    foreach (string metric in MetricNames)
    {
        float value = -1;
        PropertyInfo info;
        if ((info =
            face.Expressions.GetType().GetProperty(NameMappings(metric))
            ) != null)
        {
            value = (float)info.GetValue(face.Expressions,
                null);
        }
        else if ((info =
            face.Emotions.GetType().GetProperty(NameMappings(metric))
            ) != null)
        {
            value = (float)info.GetValue(face.Emotions,
                null);
            switch (metric)
            {
                case "Joy" :
                    Joy += value;
                    break;
                case "Sadness":
                    Sadness += value;
                    break;
                case "Anger":
                    Anger += value;
                    break;
                case "Disgust":
                    Disgust += value;
                    break;
                case "Surprise":
                    Surprise += value;
                    break;
                case "Fear":
                    Fear += value;
                    break;
                default:
                    break;
            }
        }
    }
}
}

```

Ukážka 7 Zbieranie dát pre vyhodnotenie štatistických údajov

OnRender() je metóda, ktorá sa stará o vykresľovanie vrstvy „canvas“ zobrazenej na popredí snímok vyhotovených z webkamery. Vykresľuje názvy a hodnoty jednotlivých klasifikovaných emócií prostredníctvom lišty progresu, vykresľuje jednotlivé črty tváre a obdĺžnik okolo detegovanej tváre.

```

protected override void OnRender(System.Windows.Media.DrawingContext dc)
{
    //For each face
    foreach (KeyValuePair<int, Affdex.Face> pair in Faces)
    {
        Affdex.Face face = pair.Value;
        var featurePoints = face.FeaturePoints;
        //Calculate bounding box corners coordinates.
        System.Windows.Point tl = new System.Windows.Point(featurePoints.Min(r =>
            r.X) * XScale,
            featurePoints.Min(r => r.Y) * YScale);
        System.Windows.Point br = new System.Windows.Point(featurePoints.Max(r =>
            r.X) * XScale,
            featurePoints.Max(r => r.Y) * YScale);

        System.Windows.Point bl = new System.Windows.Point(tl.X, br.Y);

        //Draw Points
        if (DrawPoints)
        {
            foreach (var point in featurePoints)
            {
                dc.DrawEllipse(pointBrush, null, new System.Windows.Point(point.X *
                    XScale, point.Y * YScale), fpRadius, fpRadius);
            }
        }
        //Draw BoundingBox
        dc.DrawRectangle(null, boundingPen, new System.Windows.Rect(tl, br));
    }
    //Draw Metrics
    if (DrawMetrics)
    {
        double padding = (bl.Y - tl.Y) / MetricNames.Count;
        double startY = tl.Y - padding;
        foreach (string metric in MetricNames)
        {
            double width = maxTxtWidth;
            double height = maxTxtHeight;
            float value = -1;
            PropertyInfo info;
            if ((info = face.Expressions.GetType().GetProperty(NameMappings(metric)))
                != null)
            {
                value = (float)info.GetValue(face.Expressions, null);
            }
            else if ((info =
                face.Emotions.GetType().GetProperty(NameMappings(metric))) != null)
            {
                value = (float)info.GetValue(face.Emotions, null);
            }
            SolidColorBrush metricBrush = value > 0 ? pozMetricBrush :
                negMetricBrush;
            value = Math.Abs(value);
            SolidColorBrush txtBrush = value > 1 ? blackBrush : boundingBrush;
            double x = tl.X - width - margin;
            double y = startY += padding;
            double valBarWidth = width * (value / 100);
            if (value > 1) dc.DrawRectangle(null, boundingPen, new
                System.Windows.Rect(x, y, width, height));
            dc.DrawRectangle(metricBrush, null, new System.Windows.Rect(x, y,
                valBarWidth, height));
        }
    }
}

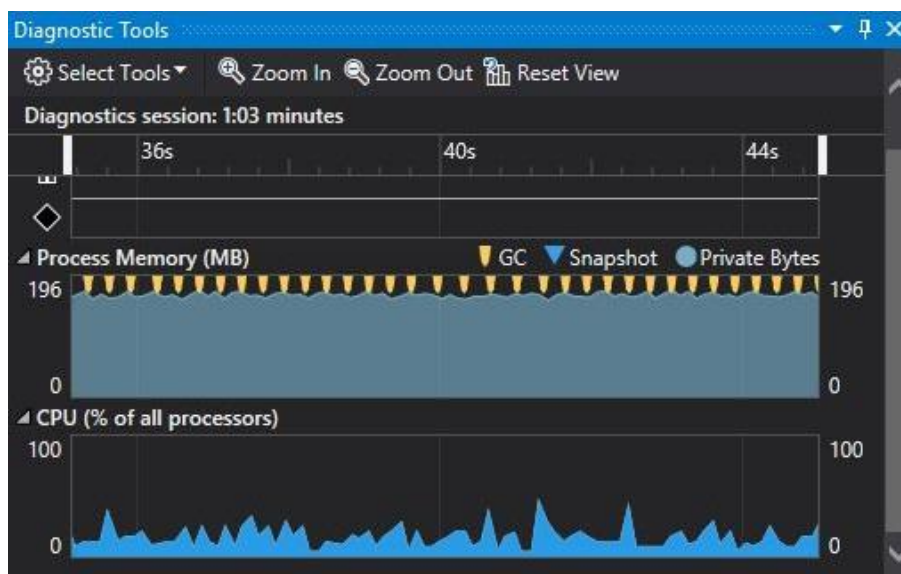
```

```
FormattedText metricFTScaled = new
FormattedText((String)upperConverter.Convert(metric, null, null, null),
System.Globalization.CultureInfo.CurrentCulture,
System.Windows.FlowDirection.LeftToRight,
metricTypeFace, metricFontSize * width / maxTxtWidth, txtBrush);
dc.DrawText(metricFTScaled, new System.Windows.Point(x, y));
}
}
}
}
```

Ukážka 8 Renderovanie klasifikácie emócie

5.5 Zhodnotenie dosiahnutých výsledkov

Podarilo sa nám naplniť ciele, ktoré sme si stanovili v 2 CIELE DIPLOMOVEJ PRÁCE. Aplikácia v reálnom čase vyhodnocuje emocionálny stav používateľa na základe vstupu z webkamery. V porovnaní s prvým riešením má systém oveľa nižšie nároky na výpočtový výkon. Na obrázku nižšie (Obrázok 21) uvádzame konkrétne namerané hodnoty zaťaženia systému z hľadiska vytťaženia procesora a pamäte RAM použitej na chod aplikácie.



Obrázok 21 Vytťaženie systémových zdrojov počas klasifikácie emócií

Namerané hodnoty je možné porovnať s prvotným riešením pomocou Obrázok 15 Vytťaženie systémových zdrojov počas detekcie tváre.

System je schopný za bežných svetelných podmienok dosiahnuť správnu detekciu tváre na vzdialenosť 7,5 metra od snímacieho zariadenia a to pri nastavení parametra `Affdex.FaceDetectorMode` na hodnotu `SMALL_FACES`. Minimálna vzdialenosť pre úspešnú detekciu tváre je 20 cm.

Miera úspešnosti aplikácie bola hodnotená za pomoci databázy JAFE (z angličtiny Japanese Female Facial Expression), ktorá obsahuje 214 fotografií japonských modeliek, ktoré vykazujú nasledovné emócie: hnev, znechutenie, strach, šťastie, neutrálny výraz, smútok a prekvapenie (JAFE, 2005).

Testovacia sada obrázkov obsahuje 10 osôb, ktorá bola vyfotografovaná počas vykazovania konkrétnej emócie 3 až 4 krát. Testovacie prostredie (Obrázok 22) bolo zostrojené nasledovným spôsobom: oproti sebe boli vo vzdialenosti 1 meter umiestnený notebook, na ktorom bol spustený nami vytvorený softvér určený na klasifikáciu emócií. Na druhej strane bol umiestnený monitor, na ktorom boli postupne zobrazované testovacie fotografie, ktoré boli softvérovo zväčšené na 200% pôvodnej veľkosti.



Obrázok 22 Ukážka určovania miery úspešnosti

Do tabuľky bol priebežne zaznamenávaný výstup z nášho klasifikačného softvéru, ktorý percentuálne vyjadroval intenzitu klasifikovanej emócie. V prípade, ak softvér vyhodnotil inú emóciu ako bola zobrazená na testovanej fotografii, do tabuľky bola zaznamenaná nula. Na konci procesu boli vypočítané priemery pre každú vyjadrovanú emóciu zvlášť.

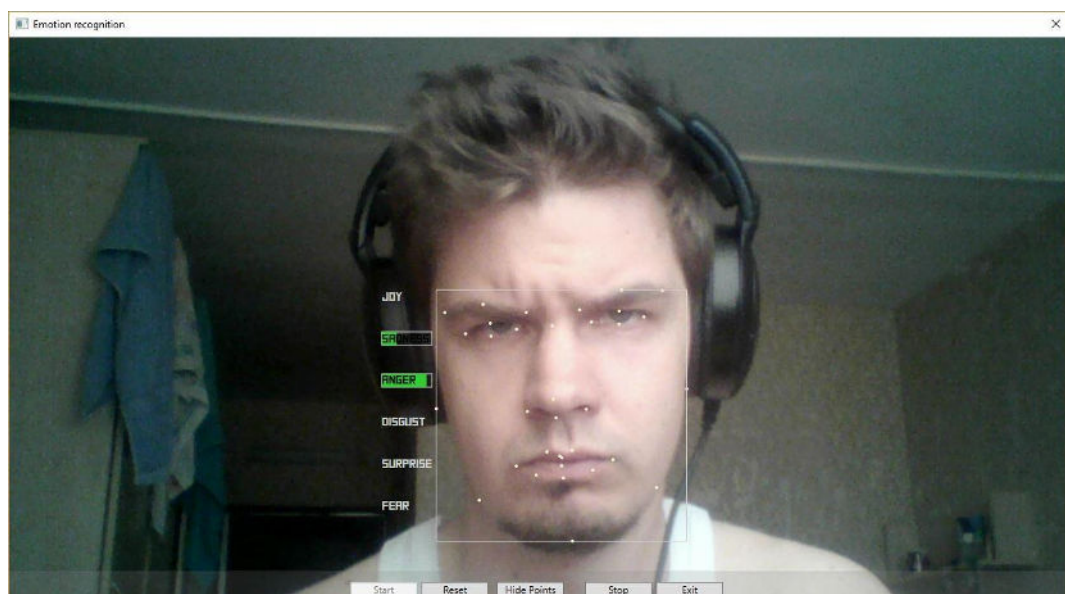
Tabuľka 3 znázorňuje mieru úspešnosti rozpoznania daných emócií pomocou nášho riešenia. Je potrebné poznamenať, že výsledky môžu byť skreslené testovacím prostredím. Naša aplikácia je navrhnutá na rozpoznávanie emócií v reálnom čase zo živého vstupu z webkamery. Počas testovania však dochádzalo ku klasifikácií emócií zo statických

obrázkov s rozlíšením 256x256 pixelov, takže naše riešenie malo v rámci tohto testovania iba obmedzený zdroj údajov na analyzovanie.

Tabuľka 3 Miera úspešnosti rozpoznávania emócií

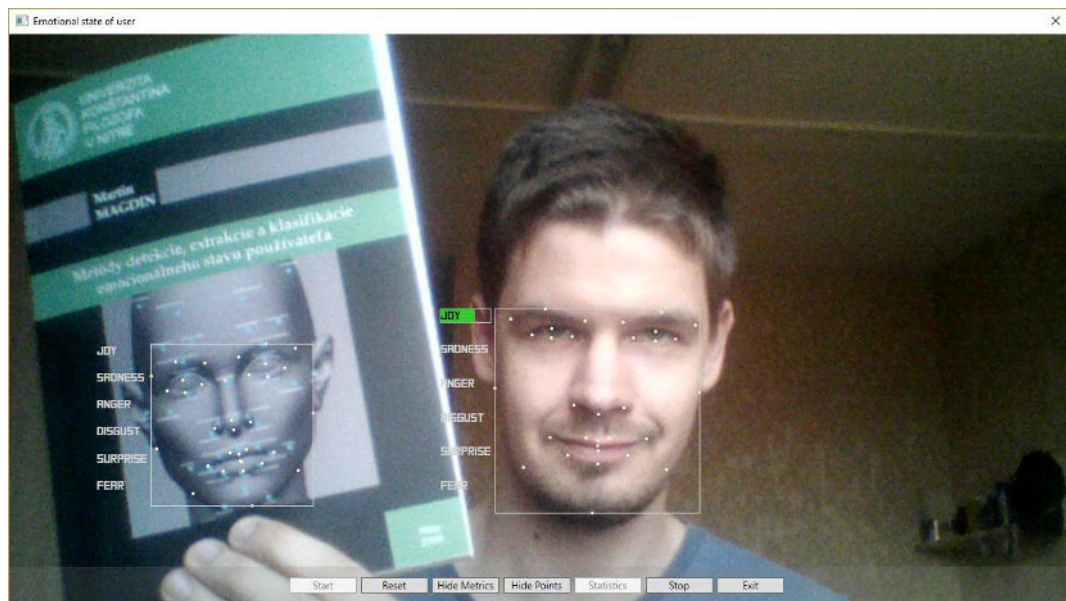
Hnev	Znechutenie	Strach	Šťastie	Neutrálny výraz	Smútok	Prekvapenie
5,33%	40,17%	7,26%	77,26%	82,67%	31,61%	46,88%

Pre porovnanie uvádzame niekoľko screenshotov testovania aplikácie v reálnych podmienkach.



Obrázok 23 Ukážka klasifikácie hnevu

Obrázok 24 znázorňuje klasifikáciu šťastia ľudskej tváre a detekciu modelu tváre z monografie.



Obrázok 24 Detekcia šťastia a tváre z obrázku

5.6 Možnosti budúceho vývoja

V budúcnosti máme v pláne rozšíriť možnosti, ktoré ponúka aktuálna štatistická funkcionálna doplnením nasledovných údajov:

- zobrazenie času trvania jednotlivých emócií,
- zobrazenie času trvania detekcie aspoň jednej tváre,
- doplnenie neutrálneho výrazu,
- doplnenie údaje o maximálnom počte detegovaných tvárí,
- grafické zobrazenie štatistických údajov.

Máme v pláne doplniť aj export štatistických dát za účelom jednoduchosti ďalšieho spracovania.

Po dokončení spomenutých úprav by bolo možné aplikáciu implementovať na platformu Android, prípadne vytvoriť webový plug-in s rovnakou funkcionálnou.

ZÁVER

V záverečnej kapitole zhodnotíme, akým spôsobom sa nám podarilo naplniť hlavný cieľ a čiastkové ciele diplomovej práce.

Ako hlavný cieľ práce sme si stanovili navrhnutie a realizáciu systému pre rozpoznávanie a vyhodnocovanie ľudských emócií v reálnom čase na základe výrazu tváre, získaného za pomoci webovej kamery. Výsledná aplikácia mala poskytovať možnosť analyzovať zozbierané údaje. V práci sme opísali, akou cestou sa nám tento cieľ podarilo dosiahnuť.

V prvej kapitole sme opísali jednotlivé výskumy v oblasti vyhodnocovania emócií tváre od roku 1649. V tejto kapitole sme uviedli rôzne prístupy a metódy, ktoré mali najväčší vplyv na rozvoj danej oblasti.

V tretej kapitole sme uviedli teoretické východiská práce. Popísali sme metódy, ktoré je možné aplikovať v rámci detekcie tváre v obraze, extrakciu črt a samotnú klasifikáciu prežívaných emócií. V tejto kapitole sme tiež opísali knižnicu OpenCV (z angličtiny „Open Computer Vision“), ktorá na open-source platforme poskytuje verejnosti mocný nástroj na spracovanie obrazu počítačom. V samostatnej podkapitole sme opísali princíp fungovanie robustného algoritmu Viola-Jones, ktorý má veľké výhody pri spracovaní obrazu – je veľmi málo náchylný na zmenu svetelných podmienok a poskytuje vysokú spoľahlivosť.

Vo štvrtej kapitole na strane 38 sme sa zameriavali na opis prvého riešenia detekcie tváre v jazyku Java. Uviedli sme uviedli zdokumentovaný postup vypracovania úlohy, návrh používateľského rozhrania ako aj realizáciu vo vývojovom prostredí. Toto riešenie je schopné detegovať ľudské tváre v obraze, za predpokladu, že nie sú horizontálne natočené o viac ako 15 stupňov.

V poslednej kapitole sme uviedli informácie o finálnom riešení problematiky vyhodnocovania emocionálneho stavu používateľa pomocou webkamery prostredníctvom „Software Development Kit“ od spoločnosti Affectiva. V kapitole sme popísali princíp fungovania daného riešenia ako aj spôsob získania unikátnej databázy, pomocou ktorej sú emócie klasifikované. Ďalej sme opísali postup pri práci s SDK. V kapitole sme uviedli najdôležitejšie triedy a metódy, ktoré boli použité na klasifikáciu emócií v reálnom čase a priebežné zbieranie štatistických údajov. V závere kapitoly sme zhodnotili dosiahnuté výsledky.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

- Affectiva. (2017). *developer.affectiva.com*. Dostupné na Internete: Affdex SDK for Windows: http://developer.affectiva.com/v3_4_1/windows/
- Azcarte, A., Hageloh, F., Sande, K., & Valenti, R. (2005). *Automatic facial emotion recognition*. Amsterdam: Universiteit van Amsterdam.
- Bahreini, K., Nadolski, R., & Westera, W. (2014). Towards multimodal emotion recognition in e-learning environments. *Interactive Learning Environments*, 590-605.
- Baum, L. E. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*.
- BioID. (2017). *BioID face database*. Dostupné na Internete: bioid.com: <https://www.bioid.com/About/BioID-Face-Database>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Black, M., & Yacoob, Y. (1995). Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. *International Conference on Computer Vision*.
- Borenstein, G. (2012). *Kyle McDonald Explains FaceTracker*. Dostupné na Internete: Makematics: <http://makematics.com/research/facetracker/>
- Boudaren, M. Y., Monfrini, E., & Pieczynski, W. (2012). Unsupervised segmentation of random discrete data hidden with switching noise distributions,. *IEEE Signal Processing Letters*.
- Cohen, I., Sebe, N., Cozman, F. G., Cirelo, M., & Huang, T. (2003). Learning bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. *Proceedings of the IEEE Computer Society Conference on Computer Vision*.
- Cootes, T. (1998). *Active Appearance Models*. Dostupné na Internete: <http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/aam.html>
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition. *IEEE Computer Society Conference*.
- Delac, K., & Grgic, M. (2007). Face Recognition. *I-TECH Education and Publishing*.

- Delac, K., Grcic, M., & Grcic, S. (2005). Independent comparative study of PCA, ICA and LDA on the FERET data set. *Wiley Periodicals*. Dostupné na Internet: http://www.face-rec.org/algorithms/Comparisons/delac_03.pdf
- Donato, G., Bartlett, M., Hager, J., Ekman, P., & Sejnowski, T. (1999). Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Ekman, P. (1994). Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychological Bulletin*.
- Ekman, P., & Friesen, W. (1978). *Facial Action Coding System: Investigator's Guide*. Palo Alto, CA: Consulting Psychologists Press.
- Essa, I., & Pentland, A. (1997). Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Ethem, A. (2004). *Introduction to Machine Learning*.
- Fasel, B., & Luettn, J. (2003). Automatic Facial Expression Analysis: A Survey. *Pattern Recognition*.
- Friedman, J., Hastie, T., & Tibshirany, R. (2000). Additive logistic regression: a statistical view of boosting. *Ann. Statist.*
- Friesen, W., & Ekman, P. (1983). Emotional facial action coding system. San Francisco: Nepublikovaný rukopis, University of California.
- Grgic, M., & Delac, K. (2005). *face-rec.org*. Dostupné na Internet: Face recognition homepage: <http://face-rec.org/databases/>
- JAFE. (2005). Dostupné na Internet: http://www.kasrl.org/jaffe_info.html
- Lanitis, A., Taylor, C., & Cootes, T. (1995). A unified approach to coding and interpreting face images. *In International Conference on Computer Vision*.
- Lienhart, R., & Maydt, J. (2002). *An extended set of Haar-like features for rapid object detection*.
- Loconsole, C., Chiaradia, D., Bevilacqua, V., & Frisoli, A. (2014). Real-Time Emotion Recognition: An Improved Hybrid Approach for Classification Performance. *Intelligent Computing Theory*.
- Loos, H. S., Wiczorek, D., & Würtz1, R. P. (dátum neznámy). *Computer-based recognition of dysmorphic faces*. Dostupné na Internet: <http://www.nature.com/ejhg/journal/v11/n8/full/5200997a.html>
- Magdin, M. (2016). *Metódy detekcie, extrakcie a klasifikácie emocionálneho stavu používateľa*. Nitra: Univerzita Konštantína Filozofa v Nitre.

- Magdin, M., Turčáni, M., & Hudec, L. (2016). Evaluating the Emotional State of a User Using a. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- Martinez, A. (1999). Face image retrieval using HMMs. *In IEEE Workshop on Content-based Access of Images and Video Libraries*.
- Mase, K. (1991). Recognition of facial expression from optical flow. *IEICE Transactions*.
- McDuff, D., Amr, M., Mahmoud, A., Mavadati, M., Turcot, J., & Kaliouby, R. e. (2016). *AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit*. San Jose.
- McDuff, D., Kaliouby, R. E., Senechal, T., Amr, M., Cohn, J. F., & Picard, R. (2013). Affective-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected In-the-Wild.
- Ming-Hsuan, Y., Kriegman, D., & Ahuja, N. (2002). Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 34-58.
- Dostupné na Internetě:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=982883>
- Oliver, N., Pentland, A., & Lafter, F. B. (2000). A real-time face and lips tracker with facial expression recognition. *Pattern Recognition*.
- OpenCV, D. (2017). *opencv.org*. Dostupné na Internetě: Cascade Classification: Haar Feature-based Cascade Classifier for Object Detection:
http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
- Otsuka, T., & Ohya, J. (1997). A study of transformation of facial expressions based on expression recognition from temporal image sequences. *Technical report, Institute of Electronic, Information, and Communications Engineers (IEICE)*.
- Otsuka, T., & Ohya, J. (1997). Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences. *In IEEE International Conference on Image Processing*.
- Pantic, M., & Rothkrantz, L. J. (2000). Automatic Analysis of Facial Expressions: The State of Art. *IEEE Transactions on Pattern Recognition and Machine Intelligence*.
- Pieczynski, P. (2007). Multisensor triplet Markov chains and theory of evidence, *International Journal of Approximate Reasoning*.
- Prikler, F. (2016). Evaluation of emotional state of a person based on facial expression. *Perspective Technologies and Methods in MEMS Design, MEMSTECH 2016*, 161-163.

- Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. . *American Psychological Association*.
- Rosenblum, M., Yacoob, Y., & Davis, L. (1996). Human expression recognition from motion using a radial basis function network architecture. *IEEE Transactions on Neural Network*.
- Saragih, J. M., Lucey, S., & Cohn, J. F. (2010). Deformable Model Fitting by Regularized Landmark Mean-Shift. *Springer Science+Business Media*.
- Scassellati, B. (1998). Eye Finding via Face Detection for a Foevated, Active Vision System. *Nat'l Conf. Artificial Intelligence*.
- Senechal, T. M., D., & Kaliouby, R. (2015). Facial Action Unit Detection Using Active Learning and an Efficient Non-Linear Kernel Approximation. *Proc. ICCV IEEE*.
- Šochman, J. (2005). *Adaboost*.
- Štanclová, J. (2009). *Template matching, Štrukturalní rozpoznávání I*. . Praha: KSI MFF UK Praha.
- Starner, T., & Pentland, A. (1995). *Real-Time American Sign Language Visual Recognition From Video Using Hidden Markov Models*. MIT.
- Stratonovich, R. (1960). Conditional Markov Processes. *Theory of Probability and its Applications* .
- Sung, K., & Poggio, T. (1998). Example-Based Learning for View-Based Humna Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(1), 39-51 .
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *PROC CVPR IEEE*.
- Viola, P., & Jones, M. (2001). *Robust Real-time Object Detection*. Vancouver.
- Wiskott, L., Fellous, J.-M., & Krüger, N. (1997). Face Recognition by Elastic Bunch Graph Mathching. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Y.Tian, T.Kanade, & J.Cohn. (2001). Recognizing Action Units for Facial Expression Analysis. *Carnegie_Mellon University, IEEE transactions on pattern recognition and machine intelligence*.
- Yacoob, Y., & Davis, L. (1996). Recognizing human facial expressions from long image sequences using optical flow. . *IEEE Transactions on Pattern Analysis*.
- Yang, G., & Huang, T. (1994). *Human face detection in a complex background*,.

ZOZNAM PRÍLOH

Príloha A – CD obsahujúce prácu vo formáte PDF

Príloha B – zdrojový program, program vo verzii standalone, 32 a 64 bit. verzia SDK

Príloha C – screenshots