SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3873

AUTOMATSKO RASPOZNAVANJE OSMJEHA IZ SLIKE LICA

Dario Jaić

Zagreb, svibanj 2015.

Sadržaj

Popis kratica

OpenCV	Open Source Computer Vision	
SVM	Support vector machine	Algoritam vektora potpore
LBP	Local Binary Pattern	Lokalni binarni uzorak

Popis slika

Popis tablica

1. Uvod

Osmjeh je jedan od najčešćih i najprepoznatljivijih ljudskih reakcija na pozitivno emocionalno stanje. U današnjem modernom svijetu, u kojemu većina ljudi ima računalo i kameru u džepu, raspoznavanje ljudskog lica i emocija, odnosno tehnologije orijentirane naročito prema ljudima, imaju veliki potencijal u razvoju. U ovom završnom radu dotaknut će se problem detekcije osmjeha. Dva programa na različit način rješavaju ovaj problem. Jedan program na jednostavan način uspoređuje točke na licu određuje smije li se osoba ili ne. Drugi program se razlikuje po tome što je složeniji i koristi više biblioteka. Algoritam radi po principu određivanja Patcheva na licu, dobavljanja odgovarajućih podataka Local Binary Pattern (LBP), računanja histograma, te sastavljanja feature vectora za tu sliku. Potom se trenira Support Vector Machine pomoću kojega računalo "nauči" razlikovati feature vektore slika s osmjehom i slika bez osmjeha. Prvo će se opisati korištene biblioteke i algoritmi pomoću kojih su implementirana rješenja. Nakon toga će se vidjeti programska izvedba, potkrijepljena kodom i diskusija rezultata ovog završnog rada.

2. METODE

U ovom poglavlju će se objasniti teoretski dio završnog rada. Bit će objašnjeni algoritmi korišteni u implementaciji.

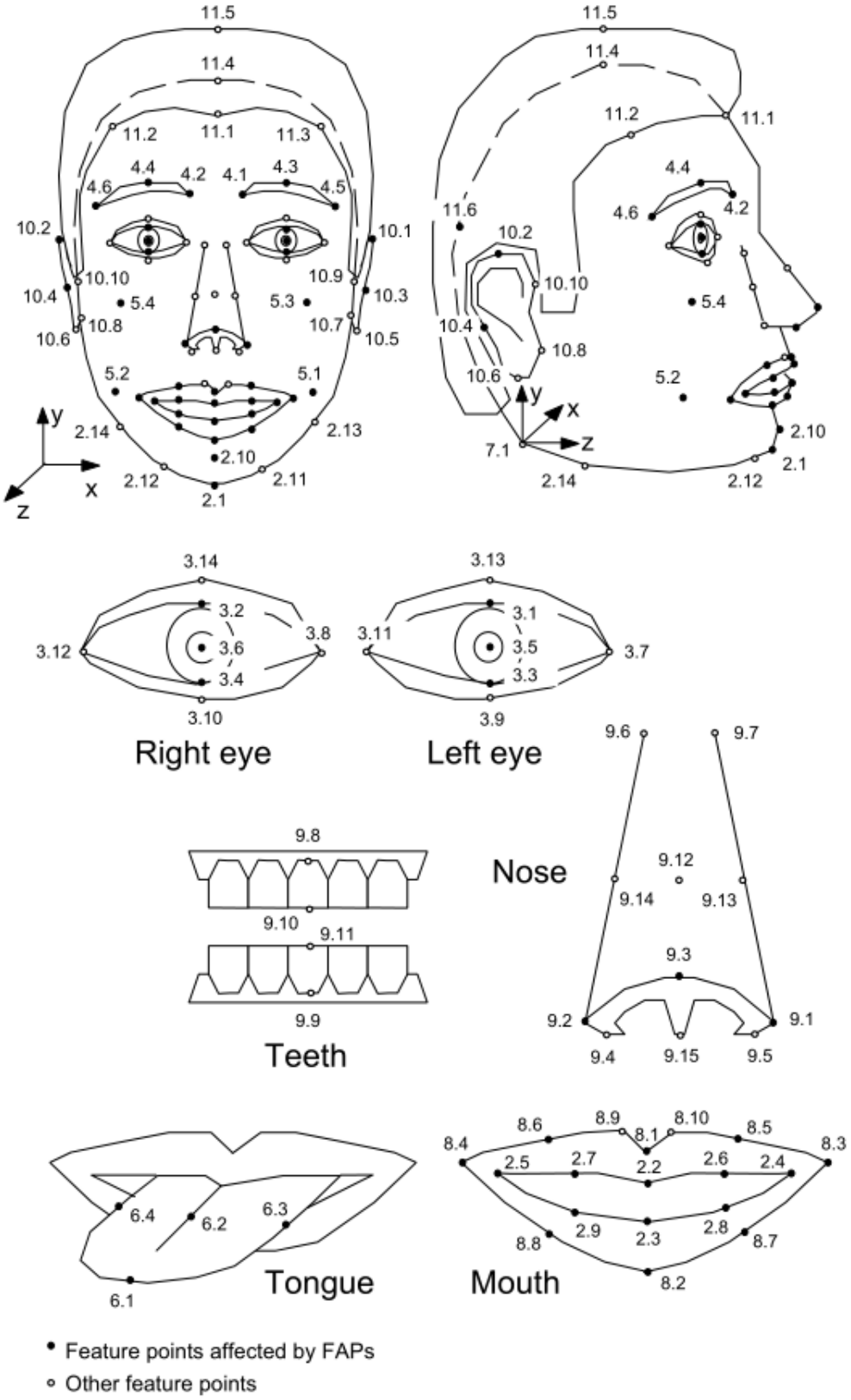
2.1 Jednostavna metoda

Prvi program detektira osmjeh na relativno jednostavan način. Feature Point je točka koja sadrži koordinate određenih točaka na modelu lica. Koordinate mogu biti dvodimenzionalne ili trodimenzionalne, ovisno o modelu. Svaka točka ima svoj tip. Na primjer, točka 8.4 predstavlja desni ekstrem usta, dok točka 3.14 gornji ekstrem desnog oka (slika 1.). Točke s desne strane lica su parni brojevi, a s lijeve neparni u skladu s označavanjem u medicinskoj anatomiji. Ako je točka dvodimenzionalna, z-os će biti nula.

U .fdp datoteci su pohranjeni feature pointovi jednog modela. Koordinate su normaliziran, tako da je potrebno pomnožiti s odgovarajućom dimenzijom slike; x-os s duljinom, odnosno y-os s visinom slike.

Program redom čita različite .fdp datoteke slika osoba. Za svaku sliku je unaprijed je određeno smije li se osoba, koja se nalazi na njoj, ili ne. Za ovu metodu su bitne četiri točke koje se mogu vidjeti na slici 1.

- 3.12 – desni ekstrem desnog oka
- 3.7 – lijevi ekstrem lijevog oka
- 8.4 – desni ekstrem usta
- 8.3 – lijevi ekstrem usta



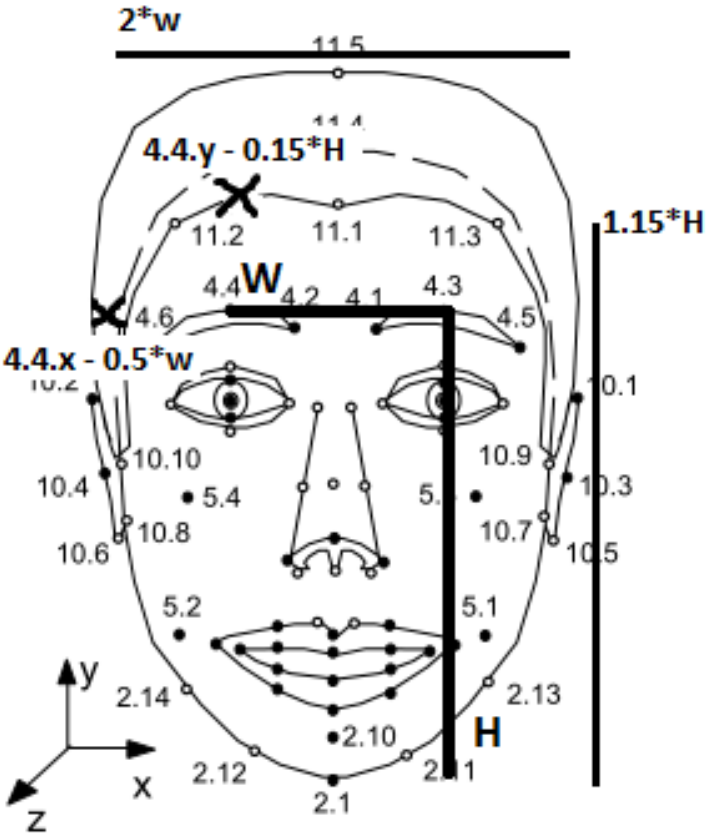
Slika 1. - Feature pointovi na licu

Također, u ovoj metodi nisu važne ordinate, nego samo apscise točke. Uspoređuje se udaljenost između točke 8.4 i 3.12, te 3.7 i 8.3. Ako je udaljenost manja od zadanog parametra, onda se osoba ne smije, ako je veća osoba se smije. Metoda se temelji na činjenici da, kada se osoba smije, usta se šire i ekstremne točke ustiju se približavaju očima. Parametar se ručno podešava u odnosu na rezultate koju su prikazani u matrici zbunjenosti (Confusion matrix). To je matrica 2x2 u kojoj prvi redak prikazuje broj slika na kojima se osoba stvarno smije, a drugi redak na kojima se ne smije. Prvi stupac prikazuje broj slika za koje program tvrdi da imaju osmjeh, dok drugi stupac prikazuje broj slika koje nemaju osmjeh.

2.2. Support Vector Machine

Prvo se učitava slika i .fdp datoteka. Dio slike se prvo odreže tako da ostane samo lice. Pravokutnik koji obuhvaća glavu, u našem slučaju, je definiran točkom koja je rub pravokutnika i duljinama stranica pravokutnika. Ta točka ima manju vrijednost koordinata od ostalih točaka pravokutnika. Algoritam je sljedeći: Koriste se 3 feature pointa koji definiraju glavu:

- 2.1 – vrh brade
- 4.4 – sredina desna obrva
- 4.3 – sredina lijeva obrva



Slika 2. - Grafički prikaz izrezivanja slike

X koordinata se računa tako da se od apscise desne obrve oduzme pola udaljenosti između obrva, također po apscisi. Tada bi se x koordinata našla na desnom rubu lica.

X = 4.4.x – 0.5 * (4.4.x – 4.3.x)

Ukupna širina lica, odnosno jedna dužina stranice pravokutnika se računa tako da se udaljenost između obrvi pomnoži s 2.

Width = 2 * (4.4.x – 4.3.x)

Y koordinata se računa tako da razlika između brade i bilo koje obrve, po ordinati, pomnoži s 0.15 i oduzme od ordinate jedne obrve.

Y = 4.4.y - 0.15 * (2.1.y – 4.3.y)

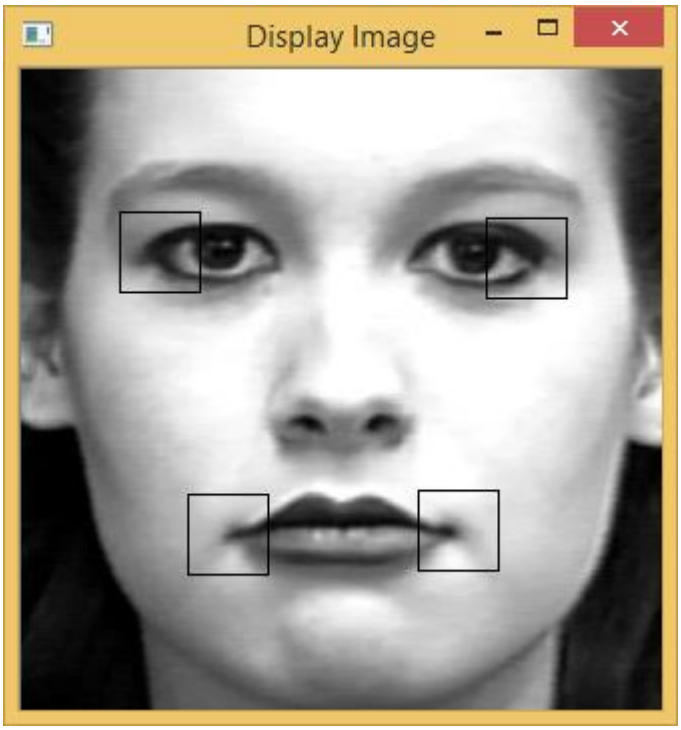
Ukupna dužina lica, odnosno dužina druge stranice pravokutnika se računa tako da se udaljenost između brade i obrve pomnoži s 1.15

Height = 1.15 * (2.1.y – 4.3.y)

S obzirom da su koordinate u .fdp datoteci normalizirane, treba ih pomnožiti još i s odgovarajućom dimenzijom slike.

Slika 3. - Neobrađena slika

Nakon što je lice izrezano sa slike, iste 4 točke se učitavaju kao i u prvom jednostavnijem algoritmu. Njihove koordinate je također potrebno promijeniti. Kako je koordinata normalizirana prvo se množi s odgovarajućom dimenzijom, te oduzima X ili Y koordinata pravokutnika i dijeli s dužinom slike, ovisno o dimenziji.



Slika 4. - Slika nakon izrezivanja lica i označavanja

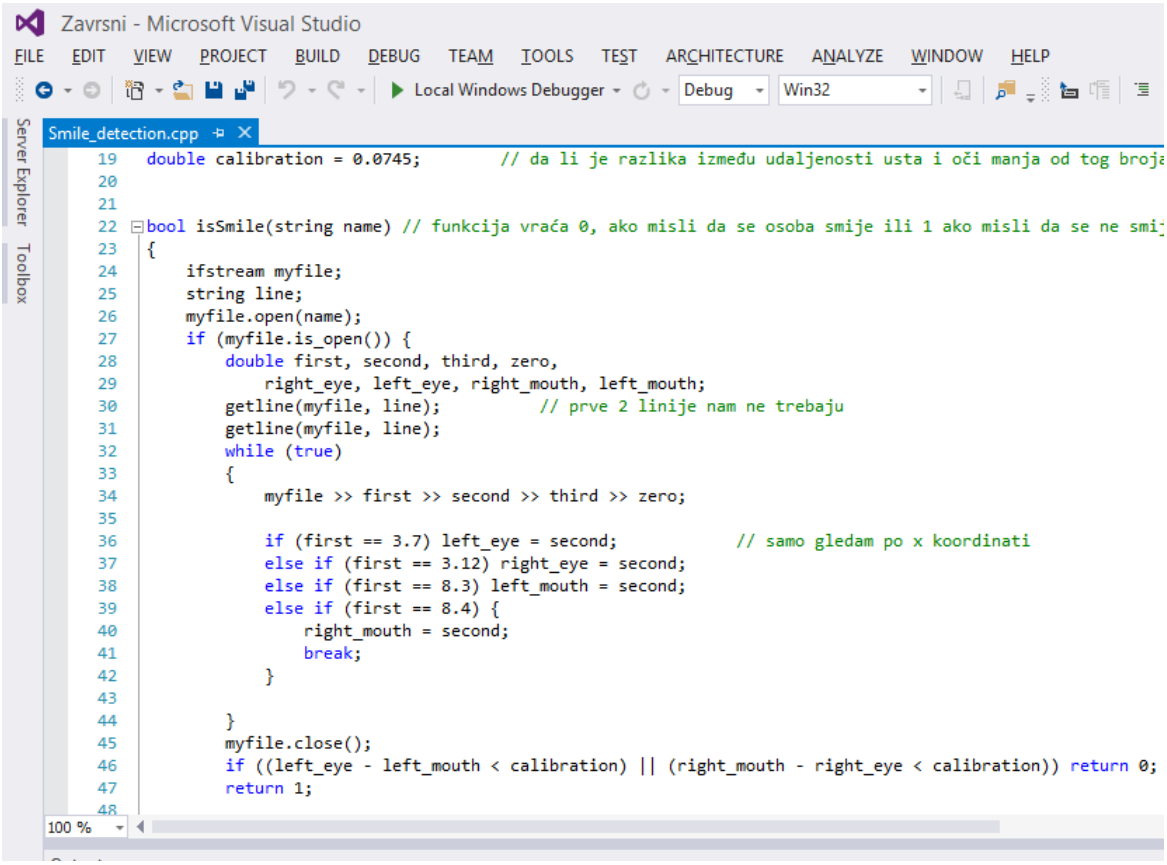
Nakon toga se definira patch (komad slike) na licu. Zadaju se dimenzije i broj kvadrata na koji je podijeljen patch. Za taj patch se generira Local Binary Pattern (LBP). Svaki pixel u patchu uspoređuje se sa vrijednošću osam susjednih pixela i tamo gdje je veća vrijednost pohranjena je 1, inače 0. Tako svaka točka ima 8-bitnu vrijednost zbog osam točaka oko sebe i prema tome se radi histogram. Histogram raspoređuje intenzitet gradijenta slike. Time je napravljen feature vector koji može pomoći u treniranju Support vector machine (SVM).
SVM je model s mogućnošću učenja i prepoznavanja uzoraka. Prilikom učenja koristi se više slika koje su podijeljene u 3 skupine. Skupina za učenje, potvrdu (validation) i testiranje. U svakoj skupini imamo slike na kojima se ljudi smiju i na kojime se ne smiju. Na temelju skupine za učenje se trenira model tako da feature vectorima na kojima se ljudi smiju pridružuje jedna vrijednost, a na onima gdje se ljudi ne smiju pridružuje druga vrijednost. Taj model se provjerava skupom za potvrdu gdje se finije podešavaju neki parametri modela. Kada su se podesili parametri novi model se ispočetka trenira sa tim novim parametrima i njega se testira sa testnom skupinom slika.

3. IMPLEMENTACIJA

Ovdje će biti prikazan programski kod i biblioteke koje su se koristile u izradi programa

3.1 Jednostavna metoda

Jednostavna metoda koristi boost biblioteku za učitavanje .fdp datoteka. U stringu je pohranjena lokacija direktorija u kojemu su podaci. Program prolazi po svim datotekama u tom direktoriju i uz pomoć regularnih izraza saznaje koje su od tih datoteka .fdp. Datoteke koje u svom imenu imaju „HA“ označavaju sliku na kojoj se osoba smije. Ime i lokacija datoteke su poslani kao argument funkciji isSmile. Funkcija otvara datoteku i iz nje čita redak po redak podatke. Prva dva su suvišna, pa ih se preskače. Ako je prvi broj jednak onim točkama koje trebaju u našem algoritmu uzet ćemo iz tog retka vrijednost na apscisi. Vrhunac programa je if naredba u kojoj se ustanovi da li razlika manja od varijable calibration. Funkcija vraća 0, ako se osoba smije, vraća 1 ako se osoba ne smije. To je tako zato što u matrici zbunjenosti na lokaciji (0, 0) se nalazi dobro pogođena slika s osmjehom. Na povratku iz funkcije isSmile funkcija regex_match iz biblioteke boost pomoću regularnog izraza provjerava da li je slika s osmjehom ili nije. Rezultati svake slike se spremaju u matricu koja se ispisuje na zaslon. Također se u datoteku sprema postotak pogođenosti slika.



Slika 5. - Funkcija isSmile

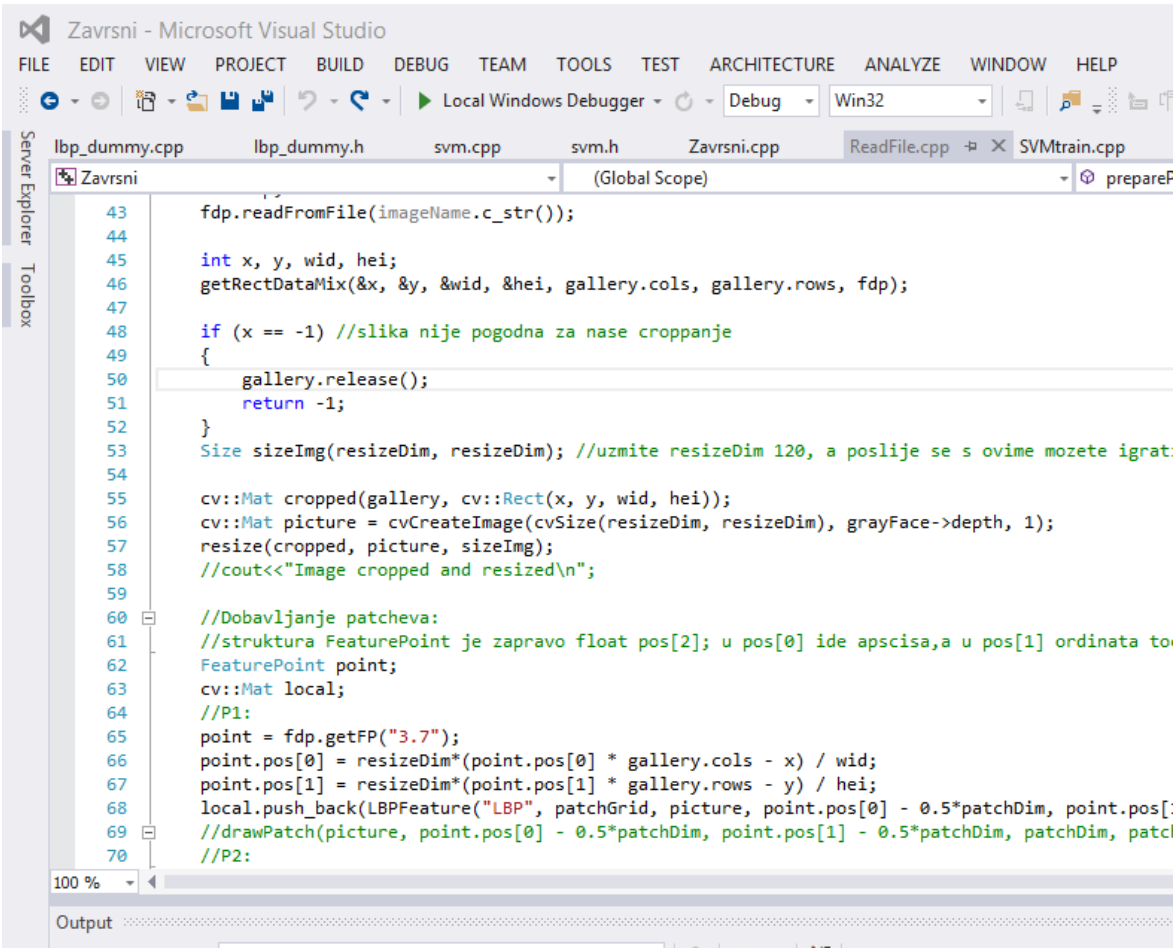
3.2. Support Vector Machine

Program sadrži biblioteke openCV-a, libSVM-a i strukture podataka iz Visage|SDK. Još sadrži i lbp_dummy.cpp i lbp.cpp datoteku s funkcijama za izradu Local Binary Patterna. OpenCV (Computer Vision) je open source biblioteka za računalni vid i strojno učenje. cv::Mat je jezgrena varijabla ove biblioteke i u nju se učitavaju podaci o slikama. OpenCV biblioteka sadrži i kod za izradu SVM-a, ali zbog iskoristivosti starog koda korišten je libSVM. Već postoje funkcije koje imaju libSVM funkcije u sebi. Od Visage|SDK koriste se struktura podataka Feature Point i klasa FDP. Klasa FDP ima u sebi korisnu funkciju readFromFile koja automatski pročita i pohrani podatke iz .fdp datoteke.
Program je napisan u četiri source datoteke:

- Zavrsni.cpp – datoteka s main funkcijom
- ReadFile.cpp – funkcija za učitavanje slike i izrade feature vectora
- SVMtrain.cpp – funkcija za treniranje modela SVM-a
- auxFunkcije.cpp – sadrži razne funkcije koje koriste ReadFile.cpp i SVMtrain.cpp

Za razliku od jednostavnog programa, ovaj program ne koristi boost biblioteku za učitavanje slika, nego otvori datoteku u kojoj se nalazi popis lokacija i imena slika koje treba otvoriti. Funkcija preparePicture iz datoteke ReadFile.cpp prima za argumet puni put do slike od funkcije main.
PreparePicture uz pomoć klase FDP dobiva feature pointove, te koristi funkciju getRectDataMix i auxFunkcije.cpp kako bi odredila dimenzije pravokutnika koji će izrezati lice sa slike. Nakon toga se postave vrijednosti koordinata ostalim 4 feature pointovima, onima koji se koriste u jednostavnom zadatku, i za svaki se pozove funkcija LBPFeature. Postoje tri varijable koje najviše utječu na cijelu izradu Feature vectora:

- resizeDim – prilikom stvaranja nove, izrezane slike ona određuje dimenzije slike
- patchDim – određuje dimenzije Patcheva na slici
- patchGrid – svaki patch ima koordinatnu mrežu koja ga dijeli na djelove, a ova varijabla određuje koliko će ta mreža imati djelova; ako je vrijednost 2 imati će 2*2 = 4 komponente



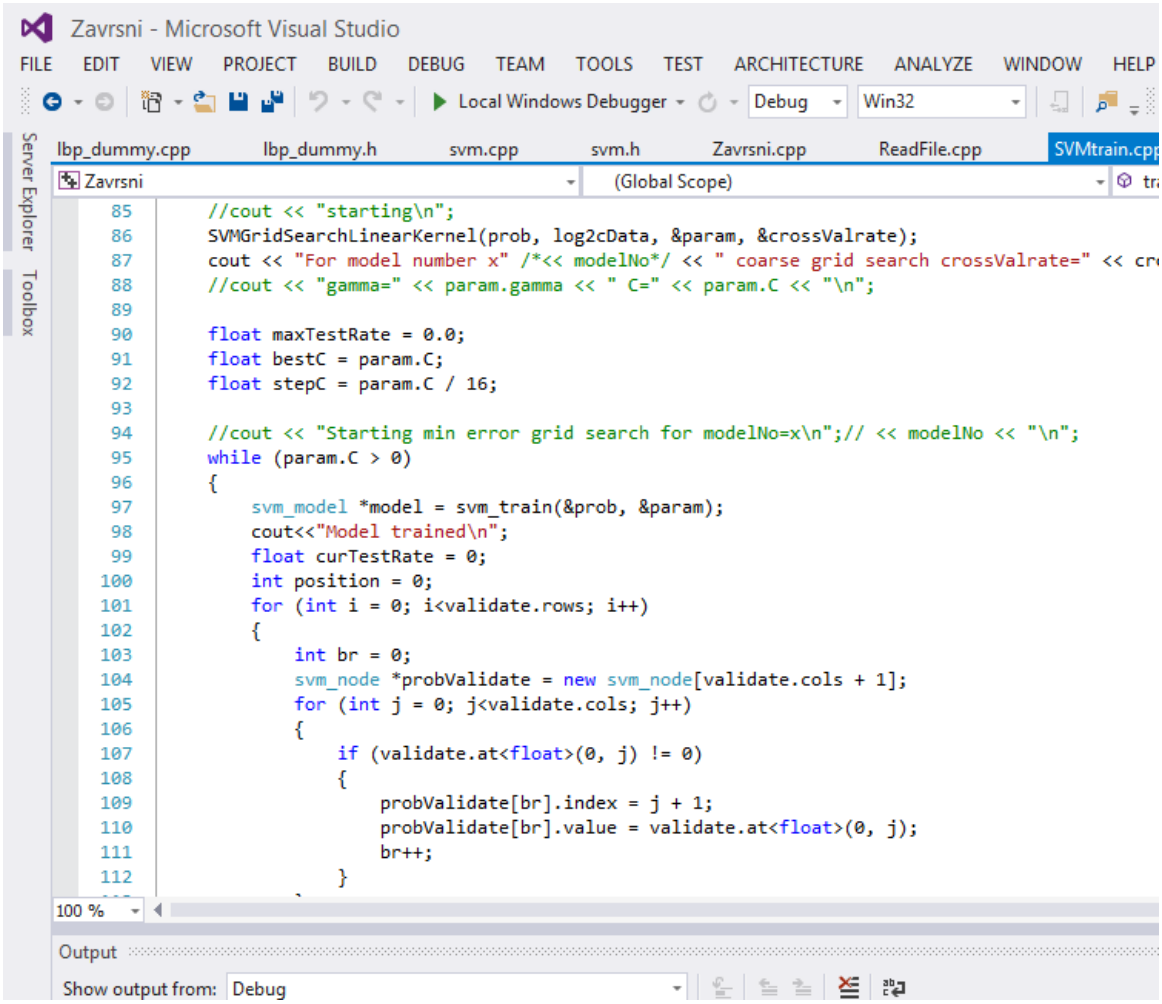
Slika 6. - Učitavanje slika

LBPFeature iz datoteke auxFunkcije.cpp sa prethodnim varijablama definira LBP i poziva funkciju histogram koja napravi histogram.

Histogram svake od 4 točke sprema se na lokalnu varijablu tipa cv::Mat. Kako su spremljene 4 točke, to je onda matrica od 4 retka. Ti retci se spoje u jedan vektor funkcijom reshape i lokalna varijable postaje Feature Vector. Lokala varijabla se, nakon toga sprema na externu listu svih feature vektora jedne vrste. Već je spomenuto da postoje tri vrste, a to su train, validate i test. Ako je spremljena slika s osmjehom poveća se brojač za slike s osmjehom za 1, inače se poveća brojač za slike bez osmjeha. Nakon što su učitane sve slike iz prve i druge vrste pokreće se funkcija trainSVM. Svaka vrsta slika je spremljena na svoju listu. Potrebno je spremiti u listu Feature vectora u tip varijable koji libSVM može čitati. Struktura tipa svm_problem, imena prob ima 3 varijable:

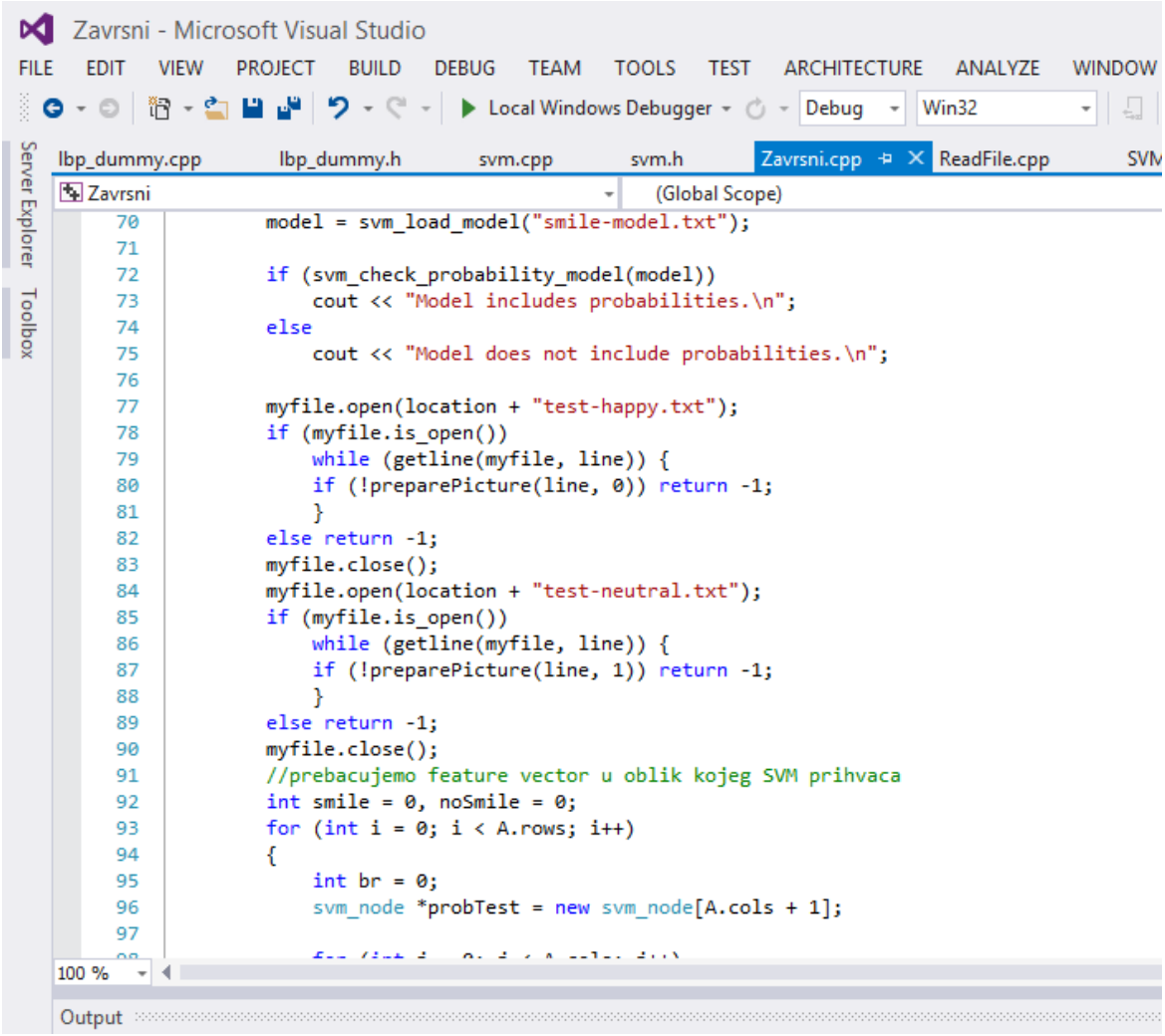
- prob.l – broj parova podataka
- prob.y – izlaz iz sustava
- prob.x – ulaz u sustav

Prob.x je tipa svm_node[prob.l] što znači da je to lista čvorova. Svaki node ima varijable index i value. Svaka slika se pridodaje svm_problem-u. Prob.l je broj slika u listi Feature vectora, Prob.y je 1, ako je slika s osmjehom, a inače -1, prob.x je vektor sa svim vrijednotima Feature vectora koje su različite od 0. Nakon što se učita lista feature vectora postavlja se svm_param, koji se sastoji od različitih parametara. Najbitiniji je parametar C koji se mijenja kako model uči. Funkcija SWMGridSearchLinearKernel uzima svm_problem, parametre i pokazivač na varijablu crossValrate.



Slika 7. - Treniranje i validacija modela

SWMGridSearchLinearKernel iz auxFunkcija.cpp će povećavati parametar C exponencijalno. Za različite vrijednosti parametra C dobit će se različita podudaranja između slika na kojima se ljudi stvarno smoju i onima za koje tako sustav tvrdi. Zapamti se vrijednost C za najbolju podudarnost i ta je postavljena svm_param objektu. S tim parametrima trenira se model sa funkcijom svm_train. Nakon što je istreniran kreće se s validacijom i sada svaka slika iz skupa za validaciju funkcijom svm_predict_probability se provjerava utreniranost modela. Sada se parametar C smanjuje logaritamski do najbolje vrijednosti obrnuto od onoga u funkciji SWMGridSEarchLinearKernel. nakon odabira najbolje vrijednosti parametra C stvara se novi model koji sada ima tu vrijednost parametra C. Taj model se sprema u datoteku „smile-model.txt“. Nad tim modelom se vrše testiranja.



Slika 8. - učitavanje i testiranje modela

4. REZULTATI

Prvo će biti prikazani rezultati jednostavnijeg programa.

Matrica zbunjenosti najboljeg rješenja:

Tablica 1. - Matrica zbunjenosti

		Predicted	
		Smile	No_smile
Actual	Smile	26	5
	No_smile	30	151

- 26 – broj točno pogođenih slika s osmjehom
- 5 – broj slika s osmjehom za koje je program tvrdi da ga nemaju
- 30 – broj slika bez osmjeha za koje je program tvrdi da ga imaju
- 151 – broj točno pogođenih slika bez osmjeha

Iz dobivenih podataka se vidi da program pogađa točno slike u 83% slučajeva, što je odličan rezultat za ovako jednostavan program. Ako je vrijednost parametra prevelika program će imati veliki postotak pogođenih slika s osmjehom, ali zato će biti puno veći broj pogriješenih slika bez osmjeha. To je zato što kad se osoba smije razlika između usta i oka je manja, nego kad se ne smije. Vrijednost parametra treba biti između te dvije razlike. Analogno tome, kada je vrijednost parametra premala većina slika je prepoznata kao ozbiljna. Dvije varijacije su se koristile prilikom testiranja. U jednoj varijaciji je uvjet, da program protumači sliku s osmjehom, bio da je razlika manja od parametra na desnoj i lijevoj strani lica, dok je u drugoj varijaciji trebala biti manja na desnoj ili lijevoj.

U tablici 1. je u prvom stupcu prikazan postotak pogođenih slika s osmjehom, u drugom bez osmjeha, a u trećem vrijednost parametra.

Tablica 2. - Rezultati prve varijante

S osmjehom [%]	Bez osmjeha [%]	Vrijednost parametra
77.4194	76.2431	0.085
74.1935	76.2431	0.080
48.3871	98.3425	0.075
93.5484	55.8011	0.090

Tablica 3. - Rezultati druge varijante

S osmjehom [%]	Bez osmjeha [%]	Vrijednost parametra
77.4194	86.7403	0.0725
77.4194	86.7403	0.0740
83.871	83.4254	0. 0745
83.871	82.8729	0. 0750
90.3226	62.9834	0. 0800
70.9677	90.0552	0. 0700

Druga varijanta je imala bolja rješenja, jer osobe se ne moraju nužno smijati simetrično, nego mogu jače na jednu stranu i zato su bolji rezultati u drugoj varijanti.

Kada je programu dana druga vrsta primjera, s trenutnim parametrom je potpuno zakazao. To je prvenstveno zato što program ne učitava sliku i uzima samo lice, nego gleda cijelu sliku. Ako lice osobe zauzima manji dio slike, onda će udaljenosti između očiju biti manje na slici, te parametar neće odgovarati i po njemu će se svi ljudi smijati. Analogno tome, ako je lice bliže, odnosno zauzima veći dio slike, lošije će se prepoznati ljudi koji se smiju. Uzimanjem jedino lica sa slike i skaliranjem udaljenosti bi se riješio taj problem.

Skup ljudi na kojima je kalibriran parametar predstavlja uzak skup ljudi. Primjeri su bile azijatske ženske osobe. Kada su ljudi više različiti rezultati su vjerojatnost pogotka je manja, jer se za sve uzima jednaka vrijednost udaljenosti. Iako bi se vrlo jednostavno moglo automatizirati traženje optimalnog parametra, to bi trebalo raditi svaki put za novu skupinu ljudi.

Rezultati kod SVM-e nisu bili zadovoljavajući. Kada bi se generirao model iz nekog razloga je uvijek za sve slike davao isti izlaz. Isprobane su gotovo sve kombinacije parametara i on uvijek davao slično. Točnost koja bi se vidjela na crossValrate je znala biti i do 93%, ali bi model zakazao.

Tablica 4. - Izlaz modela SVM

i	Smile	No Smile	resizeDim	patchDim	patchGrid
1	36	0	729	8	6
2	0	36	739	8	6
3	36	0	140	14	6
4	36	0	140	14	6
5	0	36	120	14	1
6	36	0	320	40	4
7	36	0	220	8	2
8	0	36	210	7	2
9	36	0	200	7	2
10	0	36	200	7	2

crossValrate u 7. i 8. retku je bio 93.1025% što pokazuje da je lista vektora dobro prenešena. Greška nije u listi vektora, jer nju rade gotove funkcije i dobro je prenesena u svm_problem.

5. ZAKLJUČAK

Rezultati jednostavnijeg rješenja su i više nego dobri, ali su zato od ovog kompliciranijeg nikakvi. Ili parametri nisu dobro namješteni ili je greška negdje u kodu, a s obzirom da je većina koda složenog zadatka već dobivena napisana, grešku je, izgleda, preteško naći.

6. LITERATURA

- *Boost Getting Started*, http://www.boost.org/doc/libs/1_58_0/more/getting_started/
- *Computer Vision*, http://en.wikipedia.org/wiki/Computer_vision
- *Artificial Intelligence*, http://en.wikipedia.org/wiki/Artificial_intelligence
- Hyperparameter Optimization, http://en.wikipedia.org/wiki/Hyperparameter_optimization
- libsvm.pdf*, LIBSVM: A Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- tutorials.html, *OpenCV Tutorials*, http://docs.opencv.org/doc/tutorials/tutorials.html
- *Confusion Matrix*, http://en.wikipedia.org/wiki/Confusion_matrix
- *Local Binary Patterns*, http://en.wikipedia.org/wiki/Local_binary_patterns

7. Sažetak

Naslov na hrvatskom: Automatsko raspoznavanje osmjeha iz slike lica

Naslov na engleskom: Automatic smile recognition from the facial image

Ključne riječi na hrvatskom: Algoritam vektora potpore, OpenCV, Lokalni binarni uzorak, histogram, Svojstvena točka, svojstven vektor, FDP

Ključne riječi na engleskom: Support Vector Machine (SVM), OpenCV, Local Binary Pattern(LPB), histogram, Feature Vector, Feature Point, FDP

Sažetak na hrvatskom:

Korištena su dva algoritma za raspoznavanje osmjeha iz slike lica.
Jednostavniji algoritam računa ako je udaljenost manja od zadanog parametra, onda se osoba ne smije, a ako je veća osoba se smije. Metoda se temelji na činjenici da, kada se osoba smije, usta se šire i ekstremne točke ustiju se približavaju očima.
Složeni algoritam radi po principu određivanja tzv. Patcheva na licu, dobavljanja odgovarajućih podataka Local Binary Pattern (LBN), računanja histograma (59-bin), te sastavljanja feature vectora za tu sliku. Nakon toga se trenira Support Vector Machine pomoću kojega računalo "nauči" razlikovati feature vektore slika s osmjehom i slika bez osmjeha.

Sažetak na engleskom:

Two algorithms were used for smile recognition from the facial image.
A simpler algorithm account if the distance is less than a given parameter, then the person is not smiling, and if it is higher person is smiling. The method is based on the fact that, when a person laughs, mouth spread and extreme points of mouth are approaching eyes.
A complex algorithm works on the principle of determining the so-called. Patches on the face, they supply the appropriate data Local Binary Pattern (LBN), computing a histogram (59-bin), and making Feature Vector for the image. Then the Support Vector Machine is being trained with which the computer "learn" to distinguish between feature vectors picture with a smile and without smile.