**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**POSTUPCI RAČUNALNOG RASPOZNAVANJA EMOCIJA NA OSNOVU SLIKE LICA**

**Diplomski rad**

**Josip Baketarić**

**Osijek, 2017.**

**Sadržaj**

[1 UVOD 1](#_Toc469232154)

[1.1 Zadatak rada 1](#_Toc469232155)

[2 RAČUNALNO RASPOZNAVANJE EMOCIJA 2](#_Toc469232156)

[2.1 Emocije i afekt 2](#_Toc469232157)

[2.2 Modaliteti podataka 3](#_Toc469232158)

[2.2.1 Slike lica kao izvor informacija 3](#_Toc469232159)

[2.3 Računalni postupci 4](#_Toc469232160)

[2.3.1 Klasifikacijski problem 4](#_Toc469232161)

[2.3.2 Viola-Jones algoritam 4](#_Toc469232162)

[2.3.3 Gaborovi filteri 5](#_Toc469232163)

[2.3.4 Analiza glavnih komponenti 6](#_Toc469232164)

[2.3.5 Stala odluke 7](#_Toc469232165)

[2.3.6 Nasumične šume 7](#_Toc469232166)

[2.4 Komercijalna rješenja 8](#_Toc469232167)

[3 RJEŠENJE ZA RASPOZNAVANJE EMOCIJA NA OSNOVI LICA 9](#_Toc469232168)

[3.1 Specifikacije i zahtjevi 9](#_Toc469232169)

[3.2 Korišteni alati i tehnologije 11](#_Toc469232170)

[3.2.1 OpenCV 11](#_Toc469232171)

[3.2.2 EmguCV 11](#_Toc469232172)

[3.2.3 Accord.NET 12](#_Toc469232173)

[3.2.4 AForge.NET 12](#_Toc469232174)

[3.2.5 Weka 12](#_Toc469232175)

[3.2.6 ARFF format 13](#_Toc469232176)

[3.3 Prikaz ključnih elemenata rješenja 14](#_Toc469232177)

[3.4 Prikaz izgleda, rad i uporaba rješenja 14](#_Toc469232178)

[3.5 Analiza ponašanja i performansi 14](#_Toc469232179)

[3.6 Osvrt i mogućnost unaprjeđenja 14](#_Toc469232180)

[4 ZAKLJUČAK 15](#_Toc469232181)

[5 LITERATURA 16](#_Toc469232182)

[6 SAŽETAK 18](#_Toc469232183)

[7 ŽIVOTOPIS 19](#_Toc469232184)

[8 PRILOZI 20](#_Toc469232185)

1. UVOD

„Budućnost leži u projektiranju i prodaji računala za koje ljudi ne shvaćaju da su računala.“ - Adam Osborne.

Prilikom međuljudske komunikacije, a i komunikacije između ostalih živih bića, govor ima jako važnu ulogu. Jednako važnu, ako ne i važniju ulogu prilikom komunikacije imaju emocije koje, bolje od izrečenih riječi, pokazuju trenutno emocionalno stanje osobe koje se naziva afektivno stanje. U svijetu gdje je sve veća komunikacija između računala i čovjeka ne može se zanemariti jedan tako važan aspekt ljudske prirode kao što su emocije. Kao što se sad na temelju pretraživanja sadržaja na tražilicama nude odabrane reklame, tako će se i u budućnosti na temelju afektivnog stanja nuditi određeni sadržaji.

U drugom poglavlju opisani su podatci te računalni postupci na temelju kojih se raspoznavaju emocije. Sljedeće poglavlje sadrži opis i analizu programskog rješenja kao i analizu dobivenih rezultata. Zadnje poglavlje namijenjeno je na osvrt i kratki zaključak implementiranog rješenja.

* 1. Zadatak rada

Cilj rada jest istražiti mogućnosti računalnog raspoznavanja emocija i afektivnih stanja. U teorijskom dijelu rada potrebno je opisati stanje u relevantnoj literaturi u pogledu modaliteta korištenih podataka, računalnih postupaka i dostupnih komercijalnih alata s posebnim naglaskom na uporabu slika lica kao izvora informacija. U praktičnom dijelu rada potrebno je implementirati vlastito rješenje koje omogućuje raspoznavanje emocija uporabom slika lica.

1. RAČUNALNO RASPOZNAVANJE EMOCIJA

Proučavanje emocija postoji već jako puno godina i veliki broj radova je napisano na tu temu. Jedan od temeljnih radova na spomenutu temu je Darwinov rad iz 1872. godine koji povezuje fizičke radnje, kao na primjer podizanje obrva u slučaju iznenađenja ili zbunjenosti, sa stanjem uma [1]. Raspoznavanje emocija je složen proces koji uključuje kontekst, govor tijela, kulturna obilježja, izraze lica te još par stvari [2]. U ovom radu prepoznavanje emocija je bazirano samo na izrazima lica. Računalno prepoznavanje emocija ima raznoliku upotrebu, sve od igara u kojima bi se moglo pomoću detektiranih emocija prepoznati koliko je igra zanimljiva ili dosadna, pa do različitih softvera za učenje i sustava namijenjenih za brigu o zdravlju korisnika (detekcija panike).

* 1. Emocije i afekt

Emocije su nešto što je teško definirati. Mnogo je pokušaja da se definiraju, a jedan od njih glasi da su emocije mentalna stanja [3]. Postoji više mišljenja koliko ima emocija i koje su, ali neka osnovna podjela kaže da postoje šest različitih (osnovnih) emocija. To su: sreća, tuga, ljutnja, iznenađenje, strah i gađenje. U ovom radu će se vršiti detekcija spomenutih šest emocija.

Emocije se izražavaju kroz cijelo ljudsko tijelo. Mogu se prepoznati iz tona, načina pisanja, ponašanja te izraza lica. Sve ove metode će biti spomenute i objašnjene naknadno u tekstu, a posebno metoda pomoću izraza lica na kojoj se bazira rad. Ovisno o nekoliko parametara, na licu se mogu prepoznati osnovne emocije [4]:

1. Ljutnja: Obrve su povučene prema dolje, gornji i donji kapci su povučeni prema gore, usne zategnute i malo povučene prema unutra.
2. Strah: Obrve povučene prema gore i jedna prema drugoj, gornji kapci povučeni prema gore i usta rastegnuta.
3. Gađenje: Obrve su povučene prema dolje, nos je naboran, gornja usna povučena prema gore, usne opuštene.
4. Sreća: Mišići oko očiju su stegnuti, bore oko očiju, obrazi podignuti i krajevi usana podignuti dijagonalno.
5. Tuga: Unutarnji krajevi obrva su podignuti, kapci opušteni i krajevi usana su spušteni prema dolje.
6. Iznenađenje: Obrve podignute prema gore, kapci podignuti prema gore i usta otvorena.
   1. Modaliteti podataka

Emocije su nešto što se odražava na cijelo biće i zbog toga ih je moguće prepoznati na više načina. Obično uz izraz lica dolaze i druge popratne pojave, ako je emocija snažna. Način govora se promjeni u smislu visine tona, brzine pričanja, podrhtavanja glasa i slično. Neke od bitnih značajki se pojavljuju i na pokretima tijela kao što su skupljanje ramena, tapkanje nogama, trešnja ruku i mnoge druge. Emocije se očituju i u načinu pa čak i u brzini pisanja, ali takve značajke često znaju varirati od osobe do osobe.

Slike lica kao izvor informacija

Kako bi bilo moguće prepoznati emociju to jest klasificirati izraz lica, potrebno je imati podatke za testiranje i ocjenu korištenih algoritama. Podatci korišteni u ovom radu su slike lica osoba. Po osobi ima sedam slika ukupno, šest sa prije navedenim emocijama i jedna koja je neutralna (bez emocija).

Slike koje će biti korištene prilikom stvaranja modela za klasifikaciju su slike lica koje se sastoje od 70 individua sad sedam emocija te su slikane iz različitih kutova. U svrhu ovog rada koristiti će se samo frontalne slike. Slike su preuzete sa Karolinska instituta koji se nalazi u Švedskoj [5]. Primjer slika je prikazan na slici 2.1.

|  |
| --- |
|  |
| Sl. 2.1. Primjer korištenih slika u radu, [5] |

* 1. Računalni postupci

Cilj ovog rada je naučiti računalo da prepozna emocije osobe sa slike lica. Da bi se to moglo ostvariti prvo je potrebno prikupiti odgovarajući skup podataka sa kojim će se raditi. Nakon odgovarajuće baze podataka, potrebno je napraviti pred obradu slika, kao što je na primjer promjena veličine, pretvorba slike u sliku sa sivim tonovima i slično. Iz pripremljenih slika treba izdvojiti informacije koje će biti relevantne za prepoznavanje emocija te pomoću kojih će se formirati vektor značajki.

Klasifikacijski problem

Klasifikacija uzoraka je znanstvena disciplina koja ima za zadatak razvrstavanje objekata u klase, prema [6, str.1]. Problem se nalazi u odabiru reprezentativnih značajki. Mali pokreti određenih dijelova lica određuju dali je osoba nasmijana, tužna ili nešto drugo. Ako se prilikom oblikovanja modela, prema kojem se vrši klasifikacija, koriste nepotrebne informacije (značajke dobivene iz regija pomoću kojih nije moguće detektirati emociju) vrlo je vjerojatno da će relevantne informacije biti zagušene i uspješnost klasifikacije će biti manja. Kako bi se utjecaj nebitnih ili manje bitnih regija lica smanjio, prilikom odabira značajki koriste se samo dijelovi lica gdje se nalaze oči, usta, nos i obrve. Iz svake regije potrebno je nekim metodama dobiti dovoljan broj informacija i kreirati vektor značajki koji će služiti prilikom treniranja klasifikatora. Klasifikator je algoritam koji na temelju značajki određuje klasu, ili u ovom slučaju emociju [7, str. 24].

Viola-Jones algoritam

Da bi se mogle raspoznavati emocije na licu prvo je potrebno imati sliku lica. Iako su za ovaj rad korištene slike lica, dodatno se koristi i algoritam koji sa slika detektira samo lica kako bi se izbjegao utjecaj okoline u kojoj je slika nastala. Jedan od najpoznatijih algoritama za detekciju lica na slikama je Viola-Jones algoritam. Algoritam koristi Haarove značajke koje primjenjuje na sliku i na taj način traži odgovarajući objekt, u ovom slučaju lice, usta, oči i nos [8].Haarove značajke su slike ili matrice koje na nekim mjestima imaju tamna, a na nekim mjestima svjetla područaja i svojim oblikom odgovaraju traženom objektu. Primjer se nalazi na slici 2.2.

|  |
| --- |
|  |
| Sl. 2.2. Primjer Haarove značajke, [8] |

Značajke se primjenjuju na cijelu sliku od početka do kraja i pomiče se po jedan piksel nakon primjene. Kada se prođe cijela slika značajka se povećava i sve se opet primjenjuje ponovno. Ovo nije učinkovito raditi jer je jako sporo. Kako bi se ubrzao proces traženja objekata, Viola-Jones koriste nešto što se zove integralna slika. Integralna slika je proces koji omogućuje računanje zbroja piksela unutar bilo kojeg kvadrata samo koristeći referentna polja što je prikazano na slici 2.3. [8]. Vrijednost integralne slike na lokaciji 1 je zbroj piksela u A kvadratu. Vrijednost na lokaciji 2 je A + B, na lokaciji 3 A + C i na lokaciji 4 A + B +C +D. Zbroj piksela unutar polja D se može izračunati na način da se zbroje vrijednosti na glavnoj dijagonali i oduzmu od zbroja vrijednosti na sporednoj dijagonali (1 + 4 - ( 2 + 3) ).

|  |
| --- |
|  |
| Sl. 2.1. Primjer algoritma integralne slike, [8] |

Kako bi se algoritam još ubrzao izbacuju se nepotrebne značajke pomoću Adaboost-a. Adaboost je algoritam koji izdvaja optimalne značajke iz jako velikog skupa. Nakon izdvajanja značajki, dodjeljuje im se težinska vrijednost [9]. Smatra se da je značajka prihvatljiva, ako može detektirati traženi objekt u barem pola slučajeva.

Gaborovi filteri

Gaborov filter je kompleksna sinusoidna funkcija, jednadžba 2-1, koja se primjenjuje na određenim frekvencijama i orijentacijama. Koristi se prilikom detekcije kutova, granica, klasifikacija tekstura [10]…

|  |  |  |
| --- | --- | --- |
|  |  | (2-1) |
|  |  | (2-2) |
|  |  | (2-3) |

gdje je:

-*g* – Gaborova funkcija,

-*x*, y - koordinate

-– valna dužina,

-– orijentacija,

- – Fazni pomak,

- – standardna devijacija,

- - omjer slike.

Na slici 2.2. se nalazi primjer Gaborovih filtera primijenjenih na sliku lica u četiri različite orijentacije i četiri različite valne dužine (sve ukupno 16 filtera).

|  |
| --- |
|  |
| Sl. 2.2. Gaborovi filteri primijenjeni na sliku lica |

Dvije najvažnije komponente Gaborovih filtera su valna dužina i orijentacija. Orijentacija određuje koje će se linije na slici prepoznavati i vrijednost joj se kreće od 0° do 360°. Što se tiče valne dužine, ona određuje širinu filtera koji se primjenjuje i što je širi, više linije je moguće naći, ali i moguće je i pogriješiti, ako je filter preširok. Nakon što su definirani filteri, tako zvana banka filtera, oni se primjenjuju na sliku i kao rezultat dobiva se nova slika koja ima visok intenzitet piksela gdje se poklapa sa filterom.

Analiza glavnih komponenti

Analiza glavnih komponenti (engl. *Principal component analysis*, PCA) je statistička metoda koja služi za smanjenje dimenzionalnosti podataka, detekciju ne pripadajućih vrijednosti i tako dalje [11]. Jedna od glavnih primjena PCA je smanjivanje dimenzionalnosti. PCA prebacuje podatke u manju dimenziju na način da gleda najrelevantnije podatke, a ostale zanemaruje i time se uz male gubitke informacija postigne velika ušteda u smislu brzine izračuna. PCA određuje za svaku dimenziju koliko ona sadrži kvalitetnih ili korisnih informacija i poreda ih u smislu Osnovnih komponenti. Ako bi postojao prostor od 200 dimenzija, PCA bi odredio 200 glavnih komponenti koje bi bile poredane po korisnosti. Prva glavna komponenta bi bila i najkorisnija, pa iza nje druga i tako. Na taj se način može odrediti koliko će se koristiti dimenzija i izbjeći nepotrebne podatke koje uz dodatni napor prilikom računanja mogu unijeti i pogrešku. Na slici 2.3. prikazan je dvodimenzionalni graf sa podatcima. Iz grafa se vidi da se podatci prostiru skoro kao po pravcu uz malo odstupanja. PCA prebacuje podatke u dvodimenzionalni prostor u kojemu umjesto osi, nalaze se glavne komponente. Vidi se da su podatci rasprostranjeni po Glavnoj komponenti 1, a dosta manje po glavnoj komponenti 2. Ako bi se htjelo smanjiti dimenzionalnost podataka bez puno gubitaka, može se promatrati samo projekcija podataka na glavnu komponentu 1 jer ona sadrži većinu informacija.

|  |
| --- |
|  |
| Sl. 2.3. PCA glavne komponente |

Vektori podrške

Metoda vektora podrške (engl. *Support Vector Machines*, SVM) je binarni klasifikator koji se koristi pri klasifikaciji i regresiji [6, 119. str. – 138. str]. Prostorno promatrano, SVM pokušava odvojiti značajke suprotnih klasa hiper-ravninom tako da je razmak između najbližih elemenata i hiper-ravnine najveći, slika 2.4. Kada podatci nemaju oznaku klase, nadzirano učenje nije moguće. Potrebno je dobivene podatke prvo grupirati i potom izračunati optimalnu hiper-ravninu.

|  |
| --- |
|  |
| Sl. 2.4. SVM |

U slučaju linearno odvojivih podataka, što znači da se može konstruirati hiper-ravnina koja odvaja uspješno dijeli sve podatke na dvije različite klase, hiper-ravnina se opisuje izrazom (2-4).

|  |  |  |
| --- | --- | --- |
|  |  | (2-4) |

gdje je:

-*w* – normala hiper-ravnine,

- - okomita udaljenost hiper-ravnine od ishodišta

- – podatak.

Hiper-ravnine koje odvajaju podatke i između sebe imaju najveću udaljenost su margine i mogu se opisati formulama (2-5) i (2-6), te margine se još nazivaju i vektori podrške.

|  |  |  |
| --- | --- | --- |
|  |  | (2-5) |
|  |  | (2-6) |

Ako se želi SVM koristiti i na linearno neodvojive podatke, potrebno je dozvoliti određenu grešku prilikom stvaranja hiper-ravnine. Kao i prethodno, u ovom slučaju se margine pokušavaju konstruirati sa što većim razmakom, ali pritom da je što manje pogrešno klasificiranih elemenata. Na slici 2.5. prikazana je hiper-ravnina u slučaju linearno neodvojivih podataka. Odabir hiper-ravnine se svodi na pronalaženje:

|  |  |  |
| --- | --- | --- |
|  | takav da | (2-7) |

gdje je:

-*C* – faktor pogreške,

- - mjera udaljenosti promatranog podatka od pripadajućeg vektora podrške.

|  |
| --- |
|  |
| Sl. 2.5. SVM hiper-ravnina sa linearno neodvojivim podatcima |

Iako binarni klasifikator SVM se može koristiti i prilikom više klasne klasifikacije. Dva su moguća načina kako se SVM može koristiti kao više klasni klasifikator:

1. Konstruirati n klasifikatora koji svrstava podatke u jednu od klasa ili u preostale n-1 klase. Uz rezultat klasifikacije, klasifikator vrača i mjeru sigurnosti u svoj odabir. Nakon što svi klasifikatori odrade klasifikaciju, u obzir se uzimaju samo oni koji nisu podatku pridružili klasu „ostali“. Rezultat se uzima od klasifikatora sa najvećom mjerom sigurnosti. Ovakav način se zove jedan protiv svih (engl. *One versus all*). U slučaju da svi klasifikatori odaberu klasu „ostali“, rezultat se uzima od onog sa najmanjom mjerom sigurnosti. Prikaz rada algoritma nalazi se na slici 2.6.
2. Konstruirati klasifikatora koji svaki pojedinačno određuje klasu podatka. Svaki put kad klasifikator odredi klasu toj klasi se pridodaje jedan glas. Nakon što se svi klasifikatori izvedu i postupak glasovanja završi, vrijeme je za prebrojavanje glasova. Klasa sa najvećim brojem glasova pridjeljuje se podatku. U slučaju istog broja glasova, obično se pridjeljuje klasa sa manjim indeksom. Ovakav način rada se zove jedan protiv jednog (engl. *One versus one*). Prikaz rada algoritma nalazi se na slici 2.7.

|  |
| --- |
|  |
| Sl. 2.6. SVM način rada algoritma jedan protiv svih |

|  |
| --- |
|  |
| Sl. 2.7. SVM način rada algoritma jedan protiv jednog |

* 1. Komercijalna rješenja

Prepoznavanje emocija nije ništa novo i postoji dosta gotovih rješenja za privatnu, a i komercijalnu uporabu. Neka od poznatijih rješenja su:

1. Affectiva je jako dobar alat za prepoznavanje emocija u stvarnom vremenu. Besplatna je za osobno korištenje i za tvrtke koje godišnje zarade manje od milijun dolara. Postižu veliku preciznost iz razloga jer imaju ogromnu bazu podataka za treniranje klasifikatora (četiri milijuna lica i konstantno raste). Može se koristiti u razne svrhe, ali glavna ideja je dobivanje povratne informacije od korisnika o nekom proizvodu [12].
2. EmoVu uz prepoznavanje emocija također nudi i prepoznavanje spola, godina, lica, praćenje lica i metrike raspoloženja. Ulazni podatci se šalju pomoću kamere, slike i videa. Pruža se podrška prebacivanja izračuna na grafički procesor kako bi se proces ubrzao [13].
3. Kairos također pruža uslugu prepoznavanja emocija, spola i dobi. Njihovo rješenje se može besplatno isprobati na njihovoj stranici [14].
4. Microsoftov projekt Oksford je popularan alat za prepoznavanje emocija koji radi samo na slikama lica. Prepoznaje sedam emocija i neutralno stanje. Radi na način da se pošalje slika, a kao rezultat dobije se JSON (*JavaScript Object Notation* ) format sa lokacijom lica na slici i postotkom za svaku od pruženih emocija [15].
5. RJEŠENJE ZA RASPOZNAVANJE EMOCIJA NA OSNOVI LICA

Cilj rješenja je pokušati što točnije prepoznati emocije na osnovu slika lica. Prepoznavanje emocija korisnika je dosta korisna informacija u smislu ocjenjivanja neke aplikacije i služi kao jako dobar povratni odgovor proizvođaču. Postoji dvadesetak rješenja od velikih firmi koji omogućuju korištenje njihovih rješenja u poslovne i privatne svrhe. Većina tih rješenja zahtjeva dosta računalne snage jer se koristi veliki broj klasifikatora prilikom analize lica u potrazi za akcijskim jedinicama i za svaku akcijsku jedinicu koju se želi prepoznati potrebno je imati poseban klasifikator (Ima ih 64). U ovom radu će se pokušati pojednostavniti taj proces uz što manje gubitaka i što brže te će se koristiti samo jedan klasifikator.

* 1. Specifikacije i zahtjevi

Aplikacija mora moći učitati sliku i/ili dohvatiti sliku sa kamere. Na učitanoj slici program mora pokušati pronaći lice. U slučaju pronalaska lica na slici, poziva se rutina za obradu slike i prepoznavanje emocije sa pripadajuće slike lica. Razvojno okruženje koje se koristi je *Visual Studio*, a programski jezik C#. Aplikacija se dijeli u nekoliko projekata:

* Projekt.Zajedničko– Projekt koji je referenciran u svim ostalim projektima. Koristi se za pomoćne transportne klase i sliče stvari.
* Projekt.Dohvaćanje značajki– Konzolna aplikacija. Sastoji se od tri klase. Klasa program i dvije klase za zapis značajki u datoteku u dva različita formata.

1. Program- Klasa koja služi za učitavanje podataka iz baze (baza sa slikama koja se nalazi lokalno na računalu). Poziva metodu za prepoznavanje lica na slici. Poziva metodu iz servisa koja obrađuje dohvaćeno lice i vrača vektor značajki. Nakon dohvaćenog vektora značajki, poziva metode za spremanje značajki na disk.
2. Zapis značajki– Klasa koja prima listu značajki i zapisuje ih u jednu .txt datoteku.
3. Zapis značajki arff– Klasa koja prima listu značajki i zapisuje ih u .arff formatu koji je kasnije u radu opisan.

* Projekt.Servis– Biblioteka klasa (engl. *Class library*). Glavni dio ovog projekta je klasa za obradu učitane slike, klasa za prepoznavanje lica na slici i klasa koja služi za generiranje Gaborovih filtera koji se primjenjuju na sliku.

1. Obrada slike– Klasa koja prima sliku, a kao rezultat vrača vektor značajki. Primljenoj slici se širina i visina postavlja na fiksnu dimenziju (68x80) i pretvara se u sliku sivih tonova. Slika se potom izrezuje u pet regija: gore lijevo, gore desno, dolje lijevo, dolje sredina i dolje desno. Te regije se koriste prilikom generiranja značajki. Svaka regija se šalje na obradu pomoću Gaborovih filtera koji vraća vektor značajki za tu regiju. Nakon obrade svih regija, generira se jedan vektor značajki koji reprezentira ulaznu sliku.
2. Klasifikator– Ovo je klasa koja je zadužena za pronalaženje lica na slici. Koristi se vanjski framework EmguCV i Haarov klasifikator.
3. Klasa filter– Prima ulaznu sliku. Generira šesnaest Gaborovih filtera različitih orijentacija i valnih dužina. Ti filteri se potom primjenjuju na sliku koja onda predstavlja korisne značajke. Kako bi se smanjio broj značajki, koristi se PCA. Poziva se metoda koja prima sliku i pretvara ju u 2D polje to jest matricu. Na matricu se primjenjuje PCA i onda se umjesto cijele slike, koristi samo jedan dio. Na kraju metoda vraća vektor značajki.
4. Pomoćne klase– Služe za pretvaranje slike iz *Bitmap* formata u 2D polje i obrnuto, pretvaranje slike u sliku sivih tonova, rezanje slike, promjenu veličine slike…

* Projekt.Weka– Biblioteka klasa. Služi za treniranje algoritma i klasifikaciju. Sastoji se od dvije klase: klasifikator i klasificiraj. Koristi posebnu Wekinu biblioteku kako bi se mogli koristiti klasifikatori.

1. Klasifikator- Klasa koja služi za treniranje i ocjenu klasifikatora. Korišteni klasifikator je SVM.
2. Klasificiraj- Klasa koja omogućuje klasifikaciju novih podataka. Kreira novi vektor, koji je prigodan za korištenje od strane klasifikatora, na temelju dobivenog vektora značajki. Kao rezultat vrača predviđenu klasu.

* Projekt.Form- *WindowsForms* aplikacija. Omogućuje korisniku da koristi kameru kako bi predvidio emociju i omogućuje učitavanje slike sa diska. Nakon dohvaćanja slike poziva metodu za obradu slike i metodu za klasifikaciju te na temelju dobivenih značajki ispisuje korisniku rezultat u obliku predviđene emocije.

Dijagram toka aplikacije prikazan je na slici 3.1.

|  |
| --- |
|  |
| Sl. 3.1. Dijagram toka |

* 1. Korišteni alati i tehnologije

Prilikom izrade aplikacije korišteno je nekoliko vanjskih *frameworka* kao pomoć pri izradi. To su: OpenCV, EmguCV, Accord.NET, Weka i AForge.NET. Svaki od korištenih *frameworka* je slobodan za korištenje.

OpenCV

OpenCV (*Open Source Computer Vision Library*) je *framework* otvorenog koda koja se koristi za računalni vid i strojno učenje [16]. Može se besplatno koristiti u nekomercijalne, ali i komercijalne svrhe. Sadrži više od 2500 optimiziranih algoritama koji se mogu koristiti u svrhu detekcije i prepoznavanja lica, praćenje pokreta, 3D obrada slike i videa, spajanje slika u svrhu proizvodnje slike visoke rezolucije i još mnogo toga [16]. OpenCV se koristi od osobnih korisnika, manjih tvrtki pa sve do velikih kompanija poput Google-a, Yahoo-a, Intela-a, IBM-a, Microsoft-a… Pisan je u programskom jeziku C++ i postoji mnogo *frameworka* pomoću kojih se može koristiti i u drugim programskim jezicima.

EmguCV

EmguCV je *framework* koja služi kako bi se metode iz OpenCV-a mogli koristiti u .NET kompatibilnim programskim jezicima kao što su C#, Python, VB… Pisan je isključivo u C#. Slobodan je za korištenje ako je rađena aplikacija otvorenog koda [17].

Accord.NET

Accord.NET je *framework* otvorenog koda koja se koristi za strojno učenje, obradu slike i zvuka, obradu signala, statističke primjene, prepoznavanje uzoraka… [18]… Pisan je u programskom jeziku C#. Podijeljen je u biblioteke:

* Znanstveno računanje- matematičke funkcije, statistika, strojno učenje neuronske mreže…
* Obrada slike i signala- razni filteri za slike, spajanje više slika u jednu, kreiranje integralnih slika, filteri za zvuk, detekcija lica u stvarnom vremenu…
* Biblioteke potpore- histogrami, grafovi, komponente za prikaz specifičnih slika i zvukova…

Kao i prethodne biblioteke, i ova je slobodna za korištenje.

AForge.NET

AForge.NET je *framework* otvorenog koda razvijen u C#-u. Dizajniran je za rad na područjima računalnog vida, umjetne inteligencije – obrada slike, neuronskih mreža, genetskih algoritama, strojnog učenja, robotike i slično [19]. Konstantno se održava i omogućuju nove funkcionalnosti. Postoji mnogo primjera aplikacija koje su rađene u ovom *frameworku*.

Weka

Weka je alat za rudarenje podatcima koji omogućuje korištenje i analizu performansi postupkom strojnog učenja [20]. Weka je razvijena u programskom jeziku Java. Sastoji se nekoliko načina rada: istraživačkog, pokusnog. toka podataka i naredbenog retka. Prikaz početnog korisničkog sučelja prikazan je na slici 3.2.

|  |
| --- |
|  |
| Sl. 3.2. Wekino početno sučelje |

Način rada u weki:

1. Učitavanje podataka u .arff formatu.
2. Pred obrada podataka.
3. Odabir željene akcija.
4. Klasifikacija- biranje klasifikatora i podešavanje opcija za treniranje.
5. Prepoznavanje nakupina podataka- odabir algoritma i specificiranje načina rada.
6. Učenje asocijacijskih pravila- odabir algoritma i podešavanje te prikaz rezultata.
7. Odabir atributa- algoritmi za izdvajanje relevantnih od irelevantnih atributa u učitanom skupu podataka.
8. Vizualizacija- Vizualni prikaz ovisnosti klasa o atributima
9. Prikaz rezultata. Rezultati se prikazuju pomoću matrice pogrešaka. Matrica pogrešaka je matrica koja na glavnoj dijagonali ima brojeve koji kazuju koliko je točno klasificiranih podataka za svaku od klasa, a ostali koliko je kojih klasa zamijenjeno za tu. Primjer matrice pogrešaka prikazan je na slici 3.3.

Weka omogućuje i uključivanje njezinih funkcionalnosti u neki od programskih jezika. Izvorno to je Java, ali se može prevesti da radi i u drugim jezicima poput C#-a.

|  |
| --- |
|  |
| Sl. 3.3. Matrica pogrešaka |

ARFF format

ARFF je oblik zapisa podataka koji se koristi u Weki. Sastoji se od dva djela. Zaglavlje u kojem se definiraju imena i vrsta atributa te dio gdje su podatci. Zaglavlje počinje sa ključnom riječi @RELATION nakon čega se specificira ime relacije i onda slijedi nabrajanje atributa kao što je prikazano na slici 3.4.

|  |
| --- |
|  |
| Sl. 3.4. ARFF zaglavlje |

Podatci se odvajaju od zaglavlja ključnom riječi @DATA. Atributi su odvojeni zarezom, a podatci su svaki u svom redu, slika 3.5.

|  |
| --- |
|  |
| Sl. 3.5. ARFF podatci |

Komentari se pišu tako da se na početak reda u kojemu je komentar stavi znak „%“.

* 1. Prikaz ključnih elemenata rješenja

Aplikacija se sastoji iz dva dijela. Prvi dio je za stvaranje podataka koja se koristi pri treniranju klasifikatora. Drugi dio služi za prepoznavanje emocije preko istreniranog klasifikatora na temelju dobivenih podataka. Program za stvaranje podataka radi na način da iz direktorija učitava jednu po jednu sliku. Na učitanoj slici se traži lice te ako se ne uspije naći iz nekog razloga, program preskače tu sliku i nastavlja dalje. Dio koda koji je zaslužan za traženje lica na slici je prikazan na slici 3.6.

|  |
| --- |
|  |
| Sl. 3.6. Pronalaženje lica na slici |

Pomoću imena slike određuje se pripadna emocija što se zapisuje uz pripadni vektor značajki i služi prilikom testiranja. Poziva se metoda iz projekta servisa koja prima sliku lica, a vrača pripadajući vektor značajki. Zadnje što je potrebno napraviti je zapisati značajke u neku datoteku. Ako je prvi prolazak kroz petlju, onda se prvo dodaje pripadajuće zaglavlje, ovisno o formatu zapisa. Značajke se zapisuju u dvije datoteke. Jedna je formata .tsv (engl. *Tab seperated value*), a druga je .arff. Drugi dio aplikacije, koji služi za stvarnu klasifikaciju i korištenje dobivenih podataka radi na način da se prilikom pokretanja trenira klasifikator koji će se potom koristiti. Ovaj proces traje od prilike tridesetak sekundi i pokreće se samo jednom. Nakon treniranja, moguće je koristiti kameru za dohvaćanje slika ili učitati sliku sa nekog drugog mjesta. Program koristi usluge programa servisa za dohvaćanje značajke učitane slike koje kasnije prosljeđuje klasifikatoru. Program servis je dio program koji obavlja većinu posla. Taj dio posla je odvojen u poseban program tako da se mogu koristiti njegove usluge sa različitih mjesta, a prilikom promjene implementacije samo je potrebno na jednom mjestu napraviti izmjenu i sve će raditi. Program dijeli sliku na pet regija lica i na njima primjenjuje Gaborove filtere, slika 3.7. Na svaku novu sliku se primjenjuje PCA radi smanjivanje dimenzionalnosti podataka i vraća se vektor značajki u osnovni program.

|  |
| --- |
|  |
| Sl. 3.7. Primjenjivanje Gaborovih filtera |

Zadnji dio aplikacije služi za klasifikaciju. Prostoje dvije klase od kojih jedna služi za treniranje klasifikatora, a druga je napravljena kao jedinstveni objekt (engl. *Singleton*) i služi za klasifikaciju. Jedinstveni objekt je način pisanja koda tako da se iz jedne klase može instancirati samo jedan objekt i on se koristi onda kroz cijeli program. Takav način pisanja koda je dobar za probleme tipa generatora različitih brojeva jer će se uvijek koristi isti objekt za generiranje koji će davati svaki put druge brojeve, a u slučaju da postoji više objekata tog tipa moglo bi doći da slučaja kada bi oba dali isti broj. Način funkcioniranja se nalazi na slici 3.8.

|  |
| --- |
|  |
| Sl. 3.8. Način rada jedinstvenog objekta |

Dio programa zaslužan za treniranje i testiranje klasifikatora se nalazi na slici 3.9. Koristi se SVM klasifikator iz Weke. Treniranje se odvija na način da se podatci podjele na k dijelova. U svakom prolasku se samo jedan dio koristi za testiranje, a svi ostali za trening. Takav algoritam se naziv K-unakrsna provjera (engl. *K-fld cross validation*) koji je prikazan na slici 3.10.

|  |
| --- |
|  |
| Sl. 3.9. Dio koda za treniranje klasifikatora |

|  |
| --- |
|  |
| Sl. 3.10. K-unakrsna provjera |

* 1. Prikaz izgleda, rad i uporaba rješenja

Da bi se aplikacija mogla koristiti prvo je potrebno stvoriti set vektora značajki iz odgovarajuće baze podataka. Kako bi program koji služi stvaranje baze vektora značajki radio, potrebno je bazu podataka imena „KDEF – DATABASE“ imati na radnoj površini. Svi podatci nastali spremaju se u „Data“ direktorij i biti će potrebni za rad programa. Izgleda programa koji stvara bazu vektor značajki prikazan je na slici 3.11.

|  |
| --- |
|  |
| Sl. 3.11. Izgled programa za stvaranje baze vektora značajki |

Izgleda tek pokrenutog programa nalazi se na slici 3.12.

|  |
| --- |
|  |
| Sl. 3.12. Izgled tek pokrenutog programa |

Sa slike 3.12. se vide mogućnosti programa. Moguće je učitati sliku sa diska i na njoj izvršiti klasifikaciju pomoću gumba „Učitaj sliku“. Prilikom prve klasifikacije potrebno je pričekati tridesetak sekundi da se klasifikator istrenira. Taj proces obavlja se samo jednom. Na slici 3.13. prikazan je izgled programa prilikom prve klasifikacije.

|  |
| --- |
|  |
| Sl. 3.13. Izgled programa prilikom prve klasifikacije |

Nakon što se treniranje klasifikatora završi, pojavljuje se rezultat u obliku teksta koji govori koja je emocija prepoznata na slici, slika 3.14.

|  |
| --- |
|  |
| Sl. 3.14. Rezultat klasifikacije |

Gumb „Info“ otvara novi prozor koji daje informacije o klasifikatoru. Prikazana je sveukupna preciznost klasifikacije, matrica pogrešaka i preciznost klasifikacije po pojedinoj emociji, slika 3.15.

|  |
| --- |
|  |
| Sl. 3.15. Informacije o klasifikatoru |

Zadnja opcija je klasifikacija slike sa kamere uređaja. Pomoću padajućeg izbornika se odabire kamera koja se želi koristiti i pritiskom na gumb „Pokreni“ kamera se pokreće i prikazuje sliku u sredini programa. Klasifikacija se može vršiti „ručno“ ili „automatski“ Ručna klasifikacija se odnosi na to da korisnik pritisne gumb „Prepoznaj“ i u tome trenutku se uzima trenutna slika i šalje na klasifikaciju. Automatska klasifikacije se odnosi na to da će program svaku sekundu uzimati trenutnu sliku i slati je na klasifikaciju te vračati rezultat. Primjer korištenja kamere prilikom dohvaćanja slike za klasifikaciju je prikazan na slici 3.16.

|  |
| --- |
|  |
| Sl. 3.16. klasifikacije pomoću slike sa kamere |

* 1. Analiza ponašanja i performansi

Ovdje radite ekstenzivno testiranje i analizu. Koristit ćete slike iz neke od baza koje su Vam na raspolaganju, ali i slike koje nisu iz te baze. Cilj je demonstrirati prednosti i mane Vašeg rješenja.

* 1. Osvrt i mogućnost unaprjeđenja

Kao što naslov kaže.

1. ZAKLJUČAK

Zaključiti nešto.

1. LITERATURA

[1] C. Darwin, P. Ekman, i P. Prodger, The expression of the emotions in man and animals, Oxford University Press, USA, 1998.

[2] Y. L. Tian, T. Kanade i J. F. Cohn, Facial expression analysis. In Handbook of face recognition (pp. 247-275). Springer, New York, 2005.

[3] Johnson-Laird, P. Nicholas, and K. Oatley, The language of emotions: An analysis of a semantic field, Cognition and emotion, Vol. 3, No. 2, pp. 81-123, 1989.

[4] <http://www.humintell.com/2010/06/the-seven-basic-emotions-do-you-know-them/>, pristupljeno 26.11.2016.

[5] D. Lundqvist, A. Flykt, & A. Öhman, The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9, 1998

[6]S. Theodoridis i K. Koutroumbas, Pattern Recognition (Fourth Edition), Elsevier Inc., SAD, 2009.

[7]M. N. Murty i V. S. Devi, Pattern Recognition An Algorithmic Approach, Springer, SAD, 2011.

[8] P. Viola, M. Jones, Rapid object detection using boosted cascade of simple features, In Computer Vision and Pattern Recognition, Proceedings of the 2001 IEEE Computer Society Conference, Vol. 1, pp. I-511-I-518, 2001.

[9] Y. Freund & R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting. In European conference on computational learning theory, pp. 23-37 Springer Berlin Heidelberg, 1995.

[10] S. Padmavathi, P. Prem i D. Praveenn, Locating Fabric Defects Using Gabor Filters, International Jurnal of Scientific Research Engineering & Technology, ISSN 2278 – 0882, Vol. 2 Issue 8, pp. 472-478, Studeni 2013.

[11] S. Wold, K. Esbensen, and P. Geladi, Principal component analysis, Chemometrics and intelligent laboratory systems, Elsevier Science Publishers B.V, pp. 37-52, Nizozemska, 1987.

[12] <http://www.affectiva.com/>, pristupljeno 26.11.2016.

[13] <http://emovu.com/>, pristupljeno 26.11.2016.

[14] <https://www.kairos.com/>, pristupljeno 26.11.2016.

[15] <https://www.microsoft.com/cognitive-services/>, pristupljeno 26.11.2016.

[16] <http://opencv.org/>, pristupljeno 11.12.2016.

[17] <http://www.emgu.com/>, pristupljeno 11.12.2016.

[18] <http://accord-framework.net/>, pristupljeno 11.12.2016.

[19] <http://www.aforgenet.com/framework/>, pristupljeno 11.12.2016.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann i I. H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Explorations, Vol. 11, pp. 10-18, 2009

1. SAŽETAK

Sažetak rada. Najbitnije.

**Ključne riječi:**

**ABSTRACT**

Sažetak na engleskom

**Key words:**

1. ŽIVOTOPIS

Josip Baketarić, rođen u Požegi 10. kolovoza 1993. Osnovnu školu završio u Pleternici u razdoblju od 2000. do 2008. godine. 2008. godine upisuje srednju tehničku školu u Požegi, smjer tehničar za računalstvo koju završava 2012. godine. Nakon toga upisuje Elektrotehnički fakultet u Osijeku, preddiplomski studij, smjer računarstvo kojega završava 2015. godine. 2016. godine upisuje diplomski studij na prijašnje spomenutom fakultetu, sada Fakultet elektrotehnike, računarstva i informacijskih tehnologija, smjer računarstvo, izborni blok programsko inženjerstvo kojega trenutno pohađa.

1. PRILOZI

Na CD-u:

1. Diplomski rad „Postupci računalnog raspoznavanja emocija na osnovu slike lica.docx“
2. Diplomski rad „Postupci računalnog raspoznavanja emocija na osnovu slike lica.pdf“