

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br.

Lokalizacija karakterističnih točaka lica u videu

Generalić Boris

Gulan Filip

Kopljar Damir

Miličević Andrija

Nuić Hrvoje

Šarić Fredi

Zadro Tvrtko

Zagreb, siječanj 2017.

SADRŽAJ

1. Projektni zadatak	1
1.1. Opis projektnog zadatka	1
1.2. Pregled i opis srodnih rješenja	1
1.3. Konceptualno rješenje zadatka	1
2. Postupak rješavanja zadatka	3
2.1. Pretvorba boje u nijanse sive	3
2.1.1. Prvi algoritam	3
2.1.2. Drugi algoritam	3
2.2. Drugi korak	3
3. Ispitivanje rješenja	4
3.1. Ispitna baza	4
3.2. Rezultati učenja i ispitivanja	4
3.3. Analiza rezultata	4
4. Opis programske implementacije rješenja	5
5. Zaključak	6
6. Literatura	7

1. Projektni zadatak

1.1. Opis projektnog zadatka

Lokalizacija karakterističnih točaka lica u videu ili fotografiji je tehnika koja se danas koristi u mnogim sustavima i uređajima. Susrećemo je na raznim društvenim servisima, poput *Facebook-a*, koji ju koriste za automatsko označavanje ljudi na fotografijama. Većina algoritama lokalizacije točaka lica su iznimno kompleksni i zahtijevaju veliku količinu procesorske snage i memorije, pa je težnja usmjerena na poboljšavanje tih algoritama. No razvojem i napretkom tehnologije algoritmi lokalizacije točaka lica se danas uspješno, bez velikih problema, izvode i na mobilnim uređajima koji ih koriste u raznoraznim aplikacijama poput alata za šminkanje gdje osoba može uz pomoć praćenja lica vidjeti kako bi izgledali s određenim bojama na svom licu.

Kroz ovaj projekt će se pokušati dani problem lokalizacije karakterističnih točaka lica riješiti uporabom dubokih neuronskih mreža.

1.2. TODO: Pregled i opis srodnih rješenja

Iscrpan pregled srodne literature s predloženim rješenjima. Opis postojećih ispitnih baza (linkovi na javno dostupne baze).

1.3. Konceptualno rješenje zadatka

Sam sustav za lokalizaciju karakterističnih točaka lica je podijeljen u više segmenata, tj. podsustava. Prvi segment sustava na ulaz prima sliku ili jedan vremenski okvir video isječka. Dana slika ili isječak se zatim pretvaraju u sliku sivih nijansi. Tako obrađena slika se dovodi na ulaz podsustava za izlučivanje položaja svih lica na slici te kao rezultat vraća listu u obliku: koordinate gornjeg lijevog ugla, širina i visina lica.

Tako dobivena lista se zatim iskoristi na način da se iz slike sivih nijansi izrežu

prepoznata lica i skaliraju. Pojedina skalirana lica dovede se na ulaze duboke neuron-ske mreže koja kao izlaze daje koordinate odabranih karakterističnih točaka lica. Tako dobivene točke skaliraju se u prostor početne slike ili isječka te se iscrtavaju i prikazuju korisniku sustava.

2. Postupak rješavanja zadatka

2.1. Pretvorba boje u nijanse sive

Prvi korak koji je potrebno napraviti na ulaznoj slici je pretvoriti ju u sliku sivih nijansi. Kod prikaza boja slike korištenjem aditivnog RGB (engl. *Red Green Blue*) modela postoje tri komponente: crvena, zelena i plava. Kombinacijom te tri komponente u različitim omjerima dobivamo ostale boje. Da se uočiti da podjednakom raspodjelom svih triju komponenti dobivamo boje iz sivog spektra, pa se algoritam pretvorbe u sliku sivih nijansi temelji na odabiru jedne vrijednosti iz dane tri komponente kako bi se dobila siva nijansa.

2.1.1. Prvi algoritam

Prvi algoritam je vrlo jednostavan i intuitivan. Siva nijansa pojedinog slikovnog elementa se dobiva tako da se boja slikovnog elementa rastavi na tri navedene komponente. Omjer pojedinih komponenti se zbraja te se uzima srednja vrijednost, kako je prikazano izrazom (2.1) Pretvorba boje u nijanse sive equation.2.1.1).

$$E_y = \frac{E_R + E_G + E_B}{3} \quad (2.1)$$

2.1.2. Drugi algoritam

Prvi algoritam je intuitivan i vrlo jednostavan, no u praksi ne daje najbolje rezultate. Problem leži u ljudskom oku i načinu na koji opaža boje. Ljudsko oko zelenu boju opaža puno jače nego crvenu, te crvenu opaža jače nego plavu. Stoga intuitivno slijedi da bi zelena trebala biti najzastupljenija, zatim crvena i plava. Organizacija ITU (engl. *International Telecommunication Union*) u svojoj normi ITU-R BT.709-6 [?] predlaže drugi algoritam u kojem bi crvena komponenta imala udio s 22.16%, zelena s 71.56% te plava s 7.22%, što je prikazano izrazom (??).

$$E_y = 0.2126E_R + 0.7152E_G + 0.0722E_B \quad (2.2)$$

Na slici ?? mogu se usporediti rezultati obadva izraza, gdje je jasno vidljiva prednost korištenja izraza (??).



Slika 2.1: Usporedba prvog i drugog algoritma pretvorbe boja u nijanse sive

2.2. Izjednačavanje histograma

Nakon što je slika u boji pretvorena u sliku sivih nijansi vrlo lako je moguće da kontrast na danoj slici nije najbolji. U namjeri poboljšanja kontrasta na slici te izoštravanja pojedinih elemenata koristi se izjednačavanje histograma.

Izjednačavanje histograma slike je operacija pri kojoj se slika mijenja tako da broj točaka za pojedinu nijansu sive bude približno jednoliko raspoređen. Matematički

rečeno izjednačavanje histograma implicira preslikavanje početne distribucije na širu i "uniformniju" distribuciju.

Kako je za potrebe ovog projekta korištena biblioteka *OpenCV*, za izjednačavanje histograma korištena je metoda *equalizeHist* iz navedene biblioteke. Primjer rada algoritma vidljiv je na slici ??.



Slika 2.2: Izjednačavanje histograma

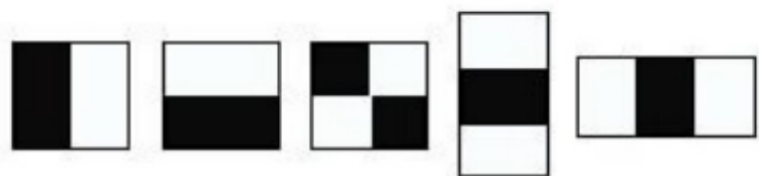
2.3. Detekcija i izlučivanje lica

Nakon što je slika u potpunosti obrađena slijedi detekcija, tj. izlučivanje pozicija lica na slici. Kako je za potrebe ovog projekta korištena biblioteka *OpenCV*, za izlučivanje pozicije lica korištene su kaskade boostanih Haarovih klasifikatora.

Metodu korištenja boostanih Haarovih klasifikatora predložili su Paul Viola i Michael Jones ne tako davne 2002. godine. To je ujedno bila prva metoda iz strojnog učenja koja postiže dobre rezultate u realnom vremenu. Iako je primarna svrha te metode prepoznavanje lica, može se iskoristiti za učenje prepoznavanja i ostalih objekata [?].

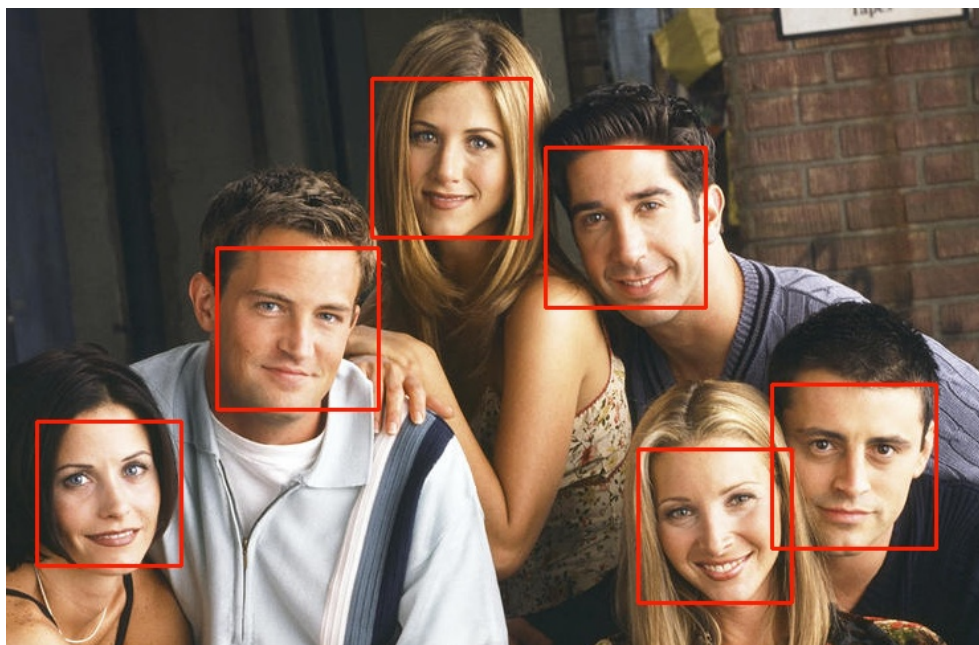
Metoda Viola-Jones umjesto direktnog rada s slikovnim elementima ulazne slike, što se pokazao kao proces s velikom složenošću, radi s određenim skupom reduciranih značajki te slike, to jest Haarovim značajkama. Haarove značajke su pravokutne značajke i računaju se kao razlika nekoliko suma slikovnih elemenata unutar različitih pravokutnih područja slike. Vrijednosti dobivene takvim računom predstavljaju prisustvo ili odsustvo nekih karakteristika slike (kao što su rubovi, kutevi i sl.) [?].

Na slici ?? prikazane su osnovne Haarove značajke. Skaliranjem, rotacijom i translacijom osnovnih značajki nastaju ostale Haarove značajke. Svaka značajka je konvolucijska maska, a vrijednost značajke, kako je već navedeno, dobiva se oduzimanjem



Slika 2.3: Osnovne Haarove značajke

sume vrijednosti slikovnih elemenata ispod bijelog područja od sume crnog područja pomnoženog s omjerom crne i bijele površine. Upravo zbog toga su te značajke vrlo osjetljive na promjenu kontrasta pa ih se interpretira kao značajke ruba ili linije [?].



Slika 2.4: Primjer rada kaskadnog Haarovog klasifikatora

CascadeClassifier razred iz biblioteke *OpenCV* u konstruktoru prima putanju do *xml* datoteke koja sadrži naučenu kaskadu Haarovih klasifikatora. Za potrebe ovog rada korištena je datoteka *haarcascade_frontalface_default* koja dolazi skupa s bibliotekom *OpenCV*. Metoda *detectMultiScale* prima na ulaz sliku sivih nijansi te vraća listu pozicija lica na slici. Primjer rada algoritma je vidljiv na slici ??.

2.4. Lokalizacija karakterističnih točaka lica

Nakon izlučivanja pojedinih lica iz slike, dana lica se režu te skaliraju na uniformne vrijednosti, u ovom slučaju na dimenziju 96×96 slikovnih elemenata. Tako skalirana

lica, tj. njihovi slikovni elementi se dovode na ulaze duboke neuronske mreže koja na svojim izlazima daje koordinate petnaest karakterističnih točaka lica: vrh nosa, lijevi i desni rub usta, sredina gornje i donje usne, centar oba oka, lijevi i desni kraj oba oka te lijevi i desni kraj obje obrve.

Primjer rada duboke neuronske mreže na lokalizaciji karakterističnih točaka lica vidljiv je na slici ??.



Slika 2.5: Lokalizacija karakterističnih točaka lica

2.4.1. TODO: Duboka neuronska mreža - Andrija ovdje ubaci teoriju

3. Ispitivanje rješenja

(do 10 stranica)

3.1. Ispitna baza

Opisati ispitnu bazu, tipove i broj različitih uzoraka u bazi te na koji su način uzorci iz baze korišteni prilikom učenja i ispitivanja rješenja projektnog zadatka.

3.2. Rezultati učenja i ispitivanja

Prikazati statističke podatke o uspješnosti rješenja prilikom učenja/ispitivanja te opisati eksperimente na temelju kojih su podaci dobiveni.

3.3. Analiza rezultata

Analizirati uzroke rezultata ispitivanja, povezati sa uzorcima u bazi i algoritmima korištenim u rješenju. Raspraviti moguća poboljšanja.

4. Opis programske implementacije rješenja

Opisati sučelje programske implementacije i način korištenja implementacije.

5. Zaključak

(do 2 stranice)

Ocijeniti uspješnost implementacije, navesti budući rad u smislu potrebnih poboljšanja.