Univerzitet u Beogradu - Elektrotehnički fakultet

Katedra za signale i sisteme

PROJEKTNI RAD

Detekcija i prepoznavanje ljudskih lica

Kandidat
Adrijana Ilić, br. indeksa 2019/3134
Suzana Mandić, br. Indeksa 2019/3120

Beograd, *avgust* 2020. godine

SADRŽAJ

1 UVOD

Sistem za prepoznavanje lica je tehnologija sposobna za identifikaciju i verifikaciju osoba na slikama i video snimcima. Postoje različiti metodi koji omogućavaju prepoznavanje lica, ali uglavnom većina njih se bazira na poređenju obilježja lica neke osobe na nekoj slici sa onima sadržanim u bazi podataka. Da bi približili sam pojam procesa prepoznavanja lica poznatijeg i kao face recognition, najbolje ga je uporediti sa procesom verifikacije. Face verification ili verifikacija lica je proces koji kao ulaz uzima sliku neke osobe i identitet (npr. ime, ID), a kao izlaz daje odgovor da li slika odgovara datom identitetu. I ovaj problem je poznatiji i kao 1:1 problem. Prepoznavanje lica ili face recognition je znatno teži problem. On podrazumijeva da imamo bazu od nekih K osoba, i da na osnovu zadate slike neke osobe, utvrdimo o kojoj osobi je tačno riječ, tj. treba da provjerimo da li ulazna slika odgovara nekoj od osoba koje imamo sačuvane u bazi podataka. Tako da se ovaj problem može posmatrati kao 1:K problem.

Ono što još otežava problem prepoznavanja lica jeste to što sistem za prepoznavanje često treba biti u stanju da prepozna neku osobu koju je vidio samo jednom, tj. često je slučaj da je u bazi podataka za neki identitet pridružena samo jedna slika te osobe. Ovaj problem je poznat i kao „one-shot learning problem“. Na sreću, postoje razvijeni različiti modeli konvolucionih neuralnih mreža (eng. Convolutional Neural Network - CNN) koje su u stanju da se izbore sa ovakvim slučajevima, i jedan od tih modela biće iskorišten u ovom radu.

Koraci koje je neophodno primjeniti da bi neko lice bilo uspješno prepoznato su sljedeći:

1. Detekcija lica i izdvajanje lica iz pozadine slike
2. Face Alignment (Poravnavanje lica)
3. Propuštanje lica kroz CNN radi izdvajanja 128 obilježja
4. Klasifikacija lica na osnovu 128 obilježja

Baza slika korištena u ovom radu je javno dostupna i poznata je kao Labeled Faces in the Wild [1]. Preuzeta je sa sajta Kaggle. Sadrži više od 13000 slika koje dolaze od preko 5000 različitih identiteta. U ovom radu nisu korišteni svi identiteti i sve slike, već je za demonstraciju procesa prepoznavanja lica iskorišten skup podataka sa 1012 i 30 identiteta.

2 Detekcija lica

Jedan od prvih koraka u prepoznavanju lica jeste izdvajanje samog lica sa slike, dakle njegovo isijecanje i samim tim odvajanje od pozadine u kojoj se ono nalazi. Da bi izdvojili lice sa neke slike, neophodno je primjeniti jedan od alata za detekciju lica, odnosno alata koji će prepoznati na kom dijelu slike se nalazi ljudsko lice, i koji to sve pikseli slike pripadaju licu.

2.1 Viola-Jones algoritam detekcije lica (HaarCascade Classifier)

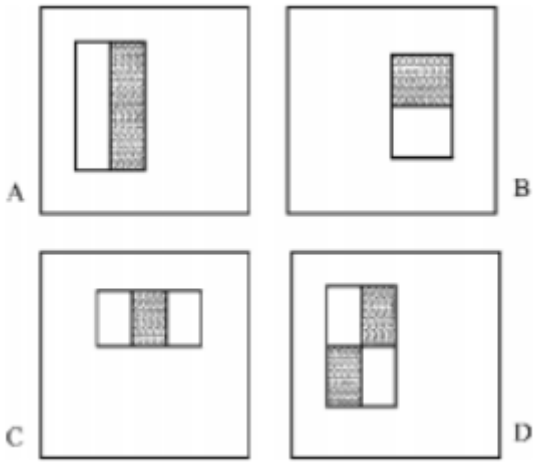
Ovo je jedan od prvih algoritama za detekciju objekata u realnom vremenu. Iako se može trenirati da detektuje više vrsta objekata, uglavnom je motivisan problemom detekcije lica. Kako bi se olakšala procedura, ovaj algoritam zahtijeva da lica budu okrenuta ka kameri, kao i da nisu nagnuta ni na jednu stranu.

Algoritam se sastoji od 4 faze:

1. Računanje odlika (Haar features)
2. Kreiranje integralnih slika
3. Treniranje AdaBoost algoritma
4. Formiranje kaskadnih klasifikatora

Računanje odlika

Ovaj algoritam ne radi direktno sa vrednostima piksela, već računa odlike i njih prosleđuje u klasifikator. Postoji više razloga zašto su autori rada odlučili za ovakav pristup. Prvi je to što korišćenjem odlika može da se enkodira ad-hoc domensko znanje koje je inače teško izvući iz konačnog broja podataka za treniranje, a drugi je to što sistemi bazirani na odlikama brže rade nego sistemi bazirani na intenzitetima piksela. Koriste se 3 vrste odlika*:* *two-rectangle*, *three-rectangle* i *four-ractangle* odlike. Two-rectangle odlika je odlika koja predstavlja razliku između sume intenziteta piksela dva pravougaona regiona na slici, i služi za detekciju ivica. Three-rectangle odlika je odlika koja se određuje tako što se najpre sračuna suma intenziteta piksela u dva spoljna pravougaona regiona, a potom oduzme od sume intenziteta piksela u centralnom regionu. Ona služi za detekciju linija. Four-rectangle odlika se određuje tako što se izračuna razlika između dijagonalnih parova pravougaonih regiona, i služi za detekciju kosih ivica. Primer odlika su prikazani na slici 2.1.1:

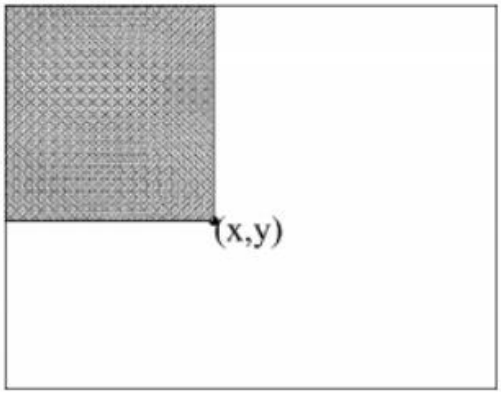


Slika 2.1.1: Two-rectangle odlike su prikazane na slikama (A) i (B). Figura (C) prikazuje three-rectangle odlike, i (D) predstavlja four-rectangle odlike.

Ukoliko uzmemo u obzir sve oguće skale i pozicije ovih odlika, i ukoliko pretpostavimo da se one računaju na prozoru veličine $24x24$ piksela, njihov broj je ogroman i iznosi 160000.

Integralna slika

Računanje 160000 ovakvih odlika je računski zahtevna operacija, pre svega jer podrazumeva mnogo operacija sumiranja. Kako bi ubrzali određivanje pravougaonih odlika, autori su koristili integralnu sliku, strukturu pomoću koje se one određuju u konstantnom vremenu i svode samo na zbir nekoliko vrednosti. Integralna slika je matrica istih dimenzija kao originalna slika, gde vrednost elememta na određenoj poziciji predstavlja sumu svih piksela u regionu iznad i levo od piksela na istoj toj poziciji u originalnoj slici (slika 2.1.2):



Slika 2.1.2: Vrednost integralne slike u tački (x,y) je suma svih piksela iznad levo.

Matematički:

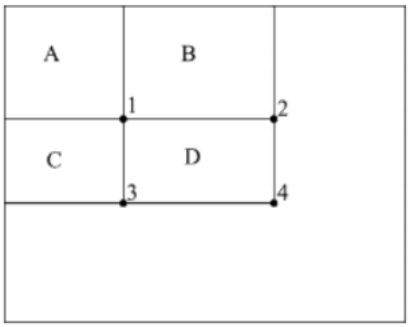
$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$$

Gde su x i y koordinate piksela u integralnoj slici, ii integralna slika, a i originalna slika. Koristeći sledeći set jednačina, integralna slika se može izračunati u jednom prolazu:

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

Pomoću integralne slike se suma svih piksela u regionu D sa slike može jednostavno izračunati kao: 4+1-(2+3) (slika 2.1.3).



Slika 2.1.3: Suma piksela unutar kvadrata D može se izračunati pomoću 4 vrednosti. Vrednost integralne slika na lokaciji 1 je suma piksela u kvadratu A. Vrednost na lokaciji 2 je A+B, na lokaciji 3 A+C, i na lokaciji 4 je A+B+C+D. Suma unutar kvadrata D je data kao 4+1-(2+3).

Pravougaone odlike su donekle primitivne u odnosu na neke alternative poput „steerable“ filtara, koji su odlični za detaljnu analizu granica, kompresiju slike, analizu teksture itd. I iako su pravougaone odlike osjetljive na prisustvo ivica, linija i drugih jednostavnih struktura, one su poprilično grube. Jedine orijentacije pravougaonih odlika koje postoje su dijagonalne, vertikalne i horizontalne. Međutim, glavna prednost pravougaonih odlika je to što se jednostavno i brzo izračunavaju, te se koriste baš zbog toga što nisu računski kompleksne.

Treniranje AdaBoost algoritma

Nakon određivanja odlika svih slika iz trening skupa, potrebno je obučiti klasifikator da prepoznaje slike lica. U ovu svrhu se može koristiti bilo koji klasifikator, ali su autori rada odlučili da koriste AdaBoost. Argumenti za ovaj izbor proizilaze iz činjenice da, iako je računanje odlika pojedinačno veoma jeftina operacija, računanje svih 160000 odlika je računski složena operacija. Koristeći AdaBoost, uz treniranje klasifikatora, moguće je ujedno i odbaciti neinformativna obeležja, i na taj način dodatno ubrzati detekciju lica (slika 2.1.4).

Slika 2.1.4: Prikaz informativnih i neinformativnih obeležja.

Kao slabi učenici AdaBoost algoritma korišteni su panjevi odlučivanja. Slab klasifikator $h(x, f, p, \theta)$ se sastoji od odlike f, praga θ i polariteta p, koji određuje smer nejdnakosti:

$$h(x, f, p, \theta) = \begin{cases} 1 \; if \; pf(x) < p\theta \\ 0 \; otherwise \end{cases}$$

gde je x podregion slike. U praksi nijedna odlika ne može izvršiti klasifikaciju sa malom greškom. Odlike koje se odaberu ranije u procesu daju grešku između 0.1 i 0.3, a odlike koje se biraju kasnije, kako zadatak postaje teži, između 0.4 i 0.5.

Table 1. The boosting algorithm for learning a query online. T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
  2. Select the best weak classifier with respect to the weighted error
     $$\epsilon_t = \min_{f,p,\theta} \sum_i w_i \, |h(x_i, f, p, \theta) - y_i| \, .$$
     See Section 3.1 for a discussion of an efficient implementation.
  3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, p_t, and θ_t are the minimizers of ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
  $$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
  where $\alpha_t = \log \frac{1}{\beta_t}$

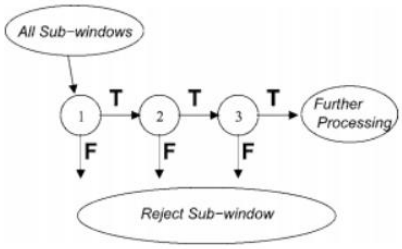
Slika 2.1.5: Pseudokod AdaBoost algoritma.

Na početku pretpostavljao da imamo trening podatke, skup pozitivnih i negativnih primera lica: (x_1, y_1),...,(x_n, y_n), gdje je $y_i = 0 \; ili \; y_i = 1$ za negativne i pozitivne primere, respektivno. Potom se inicijalizuju težine. Nakon toga imamo T iteracija, pri čemu se na početku svake iteracije normalizuju težine, onda izabere najbolji slab klasifikator, tj. onaj koji daje najmanju

otežinjenu grešku, pa se ponovo vrši klasifikacija, ažuriraju se težine i tako do konvergencije. Na kraju, kada se završe sve iteracije, formira se konačni, jak klasifikator.

Formiranje kaskadnih klasifikatora

Na ogromnoj većini podregina slike se ne nalaze lica, i ovu činjenicu autori rada koriste da bi ubrzali algoritam. Oni formiraju tzv. kaskadni klasifikator (slika 2.1.6) koji se sastoji od niza klasifikatora koji postepeno postaju sve složeniji i složeniji. Podregion se najpre prosleđuje najjednostavnijem klasifikatoru, i ako se on klasifikuje kao negatian, tu će biti kraj klasifikacije. Ukoliko se pak, ragion klasifikuje kao pozitivan, on će biti prosleđen klasifikatoru u narednom nivou gde ćeon biti opet klasifikovan itd. Na ovaj način će se potprozori slike koji definitivno ne sadrže lice, što čini veliku većinu, brzo odbaciti, dok će potprozori koji verovatnije sadrže lice biti ili odbačeni u dubljim nivoima ili će proćo sve nivoe i biti prepoznati kao lice.



Slika 2.1.6: Kaskadni klasifikator.

Nivoi u kaskadi se konstruišu koristeći AdaBoost algoritam.

Treniranje kaskadnog klasifikatora podrazumeva optimizacionu strukturu u kojoj se podešava broj nivoa, broj odlika u svakom nivou, i prag svakog nivoa, tako da se minimizira očekivani broj odlika N, pod pretpostavkom da imamo cilj stopu lažnih alarma F i stopu detekcija D. Ovdje važi:

$$F = \prod_{i=1}^{K} f_i \, , D = \prod_{i=1}^{K} d_i$$

Međutim, ovo je komplikovan problem, pa se u praksi koristi jednostavna procedura za kreiranje efikasnog klasifikatora. Korisnik bira maksimalnu prihvatljivu vrednost za f_i i minimalnu prihvatljivu vrednost za d_i. Svaki nivo kaskade se trenira pomoću AdaBoost algoritma, postepeno povećavajući broj odlika sve dok se ne postigne ciljana stopa detekcije i lažnih alarma za taj nivo. Stope se određuju testirajući klasifikator na validacionom skupu.

Sve što je neophodno da se odradi detekcija pomoću haarscascade klasifikatora, jeste učitavanje već istreniranog modela, i propuštanje željene slike kroz njega da bi se izvršila detekcija lica.

Slika 2.1.7: Detekcija lica Viola-Jones algoritmom. Od 100 lica, njih 97 je detektovano.

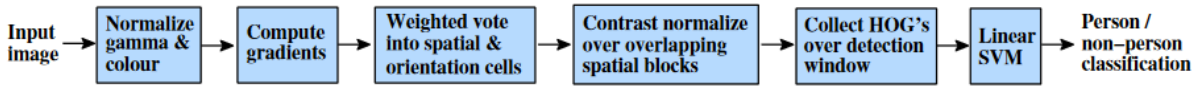
2.2 Detekcija lica pomoću Dlib biblioteke (HOG+SVM)

Drugi alat za detekciju lica je dostupan u biblioteci dlib i koristi kombinaciju HOG (Histogram of Oriented Gradient) obeležja, tj. histograma orijentisanih gradijenata i SVM-a, tj. metode nosećih vektora. Dlib za razliku od OpenCV-a je brži i precizniji, i uspeva da detektuje lica i onda kada ona nisu okrenuta pravo ka kameri, tj. kada su okrenuta u stranu ili nagnuta.

Tehnika detekcije ljudskih lica pomoću HOG obeležja objavljena je 2005-e godine i od tada je citirana više od 30000 puta. Autori ove tehnike su *Navneet Dalal* i *Bill Triggs*, a rad je poznat

pod imenom „*Histogram of Oriented Gradients for Human Detection*“ [2]. Rezultati koje je HOG pokazao na do tada najčešće korišćenom skupu slika za obuku i testiranje klasifikatora bili su gotovo bez greške, pa je formiran izazovniji *INRIA* trening/test skup slika, na kome su rezultati uglavnom bili za red veličine bolji u odnosu na ranije dostupne klasifikatore.

Tok ekstrakcije obilježja i detekcije objekata je prikazan na slici 2.2.1:



Slika 2.2.1: Tok ekstrakcije HOG obilježja i klasifikacije pomoću SVM klasifikatora.

Da bi pojam HOG obeležja bio jasan najbolje ga je raščlaniti i pojasniti šta svaka reč tog pojma predstavlja.

Gradijent u digitalnoj obradi slike govori o trendu signala intenziteta, tj. da li taj signal raste ili pada. S obzirom da je slika $2D$ signal, tako i gradijent slike predstavlja uređeni par promjene po x i y osama:

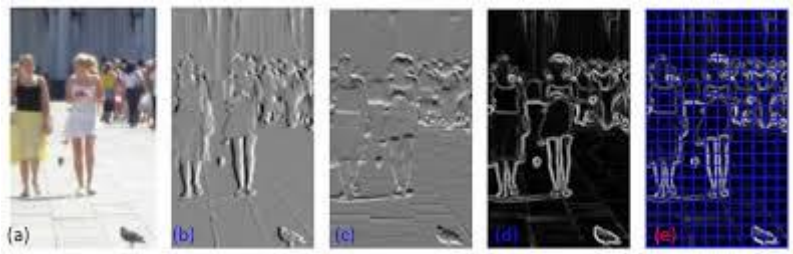
$$\nabla f(x,y) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$$

Vrijednosti gradijenta odgovaraju združenim promjenama intenziteta duž horizontalnog i vertikalnog pravca. Moduo gradijenta se računa kao:

$$G = |\nabla f| = \sqrt{{G_x}^2 + {G_y}^2}$$

Dok je ugao (pravac) gradijenta dat kao:

$$\theta = arctg\,\frac{G_y}{G_x}$$



Slika 2.2.2: Slika (b) predstavlja sliku gradijenta duž horizontalnoh pravca; slika (c) predstavlja sliku gradijenta duž vertikalnog pravca; dok je slika (d) slika modula gradijenta.

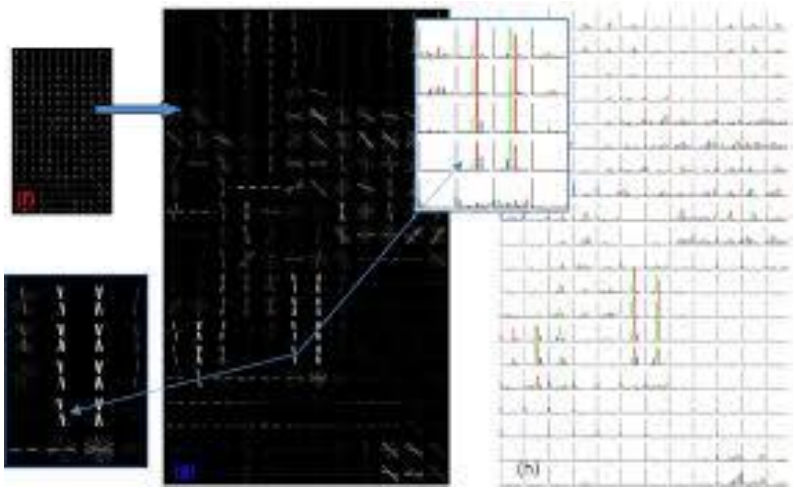
Orijentacija gradijenta u svakom pikselu je sadržana u slici uglova.

Da bi dobili HOG obilježja, potrebno je prvo diskretizovati broj mogućih vrijednosti pravaca gradijenata na slici. Prebrojavanjem koliko od kojih mogućih pravaca gradijenata na nekom dijelu slike imamo, dobijamo histogram orijentacije gradijenata tog dijela slike.

U praksi, određivanje HOG obilježja neke slike se radi tako što se slika podijeli na regione ili tzv. ćelije, pa se potom za svaku ćeliju formiraju histogrami orijentacije gradijenata, a potom se radi normalizacija po blokovima sastavljenim od više ćelija. Razlog iz kog se histogrami računaju za ćelije a ne za svaki piksel posebno je taj što bi čuvanje gradijenta za svaki pojedinačni piksel bilo previše detaljno i bespotrebno i zahtijevalo bi mnogo više memorije.

S obzirom da se za lokalizaciju objekata na slici koristi pristup višestrukih klizećih prozora (eng. Multiscale sliding windows technique), dimenzija ćelije zavisi od dimenzije prozora u kome se vrši analiza, ali zavisi i od odnosa dimenzija prozora i veličine objekta u slici čija se detekcija vrši.

Nakon što se definišu mogući pravci za gradijent, za svaki piksel se vrši i „projekcija“ gradijenta na 2 najbliža susjedna pravca. Projekcija podrazumijeva preraspodjelu vrijednosti modula gradijenta na dozvoljene, diskretne pravce i to u odnosu koji odgovara ugaonoj udaljenosti gradijenta od svakog od njih. Na ovaj način se vrši interpolacija gradijenta.

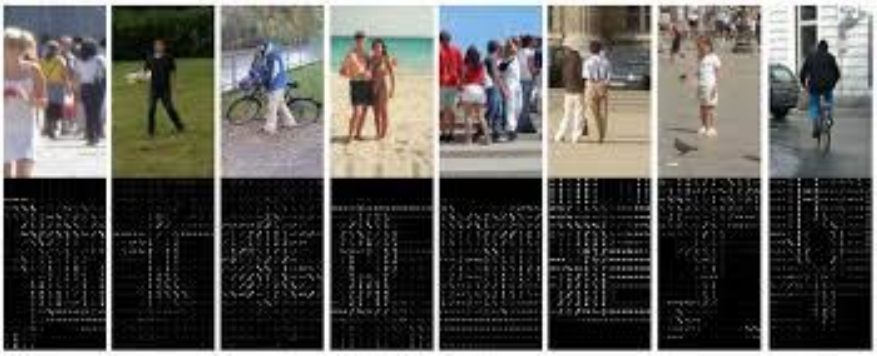


Slika 2.2.3: Histogram orijentisanih gradijenata koji odgovara slici 1.2.1(a).

U radu Dalaala i Triggsa utvrđeno je da je dovoljan broj pravaca 9 ukoliko važi:

$$\theta \epsilon [0^{\circ}, 180^{\circ}]$$

Vrijednosti histograma opisuju prisustvo dominantnih pravaca u posmatranoj ćeliji.



Slika 2.2.4: Vizualizacija HOG obilježja.

Za razliku od ostalih deskriptora HOG je invarijantan na geometrijske i svjetlosne transformacije, ali zavisi od rotacije objekata i veličine deskriptorskih blokova.

Na osnovu HOG karakteristika pozitivnih i negativnih slika, tj. slika sa licima i slika bez lica, obučava se SVM klasifikator koji nalazi optimalnu hiper-ravan kao funkciju odlučivanja.

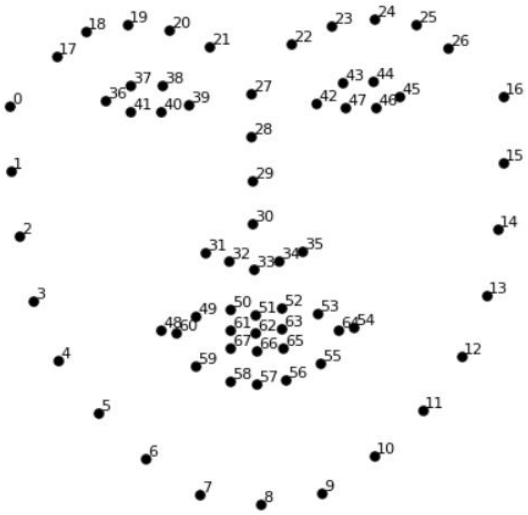
Slika 2.2.5: HOG obilježja ljudskog lica.

Slika 2.2.6: Detekcija lica pomoću HOG obilježja. Od 100 lica, njih 99 je detektovano.

3 Poravnavanje lica (Face Alignment)

Da bi omogućili lakše i preciznije prepoznavanje lica od strane klasifikatora, neophondno je da se nakon izdvanjanja lica izvrši njegovo „ispravljanje“. S obzirom da lica mogu zauzimati različite orijentacije, neophodno je „ispraviti“ lice tako da glavna obilježja lica (oči, usne, nos) nađu uvijek na istom mjestu na slici.

Ovo tzv. Ispravljanje lica je moguće odraditi pomoću algoritma koji se zove „Face Landmark Estimation“ koji je u stanju da pronađe obilježja lica na slici, ma kako god ono bilo okrenuto i orijentisano. Postoji različiti modeli estimacije obilježja lica u zavisnosti od predefinisanog broja tih obilježja koje korisnik želi da pronađe. U ovom radu je korišten model sa 68 obilježja, raspoređenih kao na slici 3.1:



Slika 3.1: Face Landmarks

Nakon što se pronađu ove tačke na slici, potrebno je rotirati, skalirati ili rašiti sliku tako da oči, usta i nos budu centrirani najbolje moguće.

4 OpenFace

Nakon izdvajanja i „ispravljanja“ lica neophodno je sliku predstaviti u obliku 128-dimenzionalne reprezentacije ili tzv. embeddinga. U takvom 128-dimenzionom prostoru Euklidska distanca direktno odgovara mjeri sličnosti lica. Da bi iz svake slike izdvojili vektor sa 128 elemenata koristi se konvoluciona neuralna mreža.

Model mreže je poznat kao nn4.small2 model u OpenFace projektu [3]. U radu je korišćena Keras implementacija ovog modela [4]. Kompletan model je definisan u model.py, a njegova grafička reprezentacija je dostupna ovdje [5].

Tokom treniranja ovakve neuralne mreže cilj je da izlaz mreže u obliku $f(x)$, gdje je x ulazna slika, bude takav da distanca između svih lica jedne iste osobe bude mala, a distanca između lica različitih osoba bude velika. Ovo se postiže pomoću funkcije trostrukog gubitka (*triplet loss function*) L. Kao što i samo ime kaže, ova funkcija se računa na osnovu 3 slike poznate i kao „*anchor image*“, „*positive image*“ i „*negative image*“. „Anchor image“ i „positive image“ predstavljaju 2 različite slike iste osobe, dok je „negative image“ slika neke druge osobe. Funkcija trostrukog gubitka glasi:

$$L = \sum_{i=1}^{m} \left[\left\| f\left(x_i^A\right) - f(x_i^P) \right\|^2 - \left\| f\left(x_i^A\right) - f(x_i^P) \right\|^2 + \alpha \right]_+$$

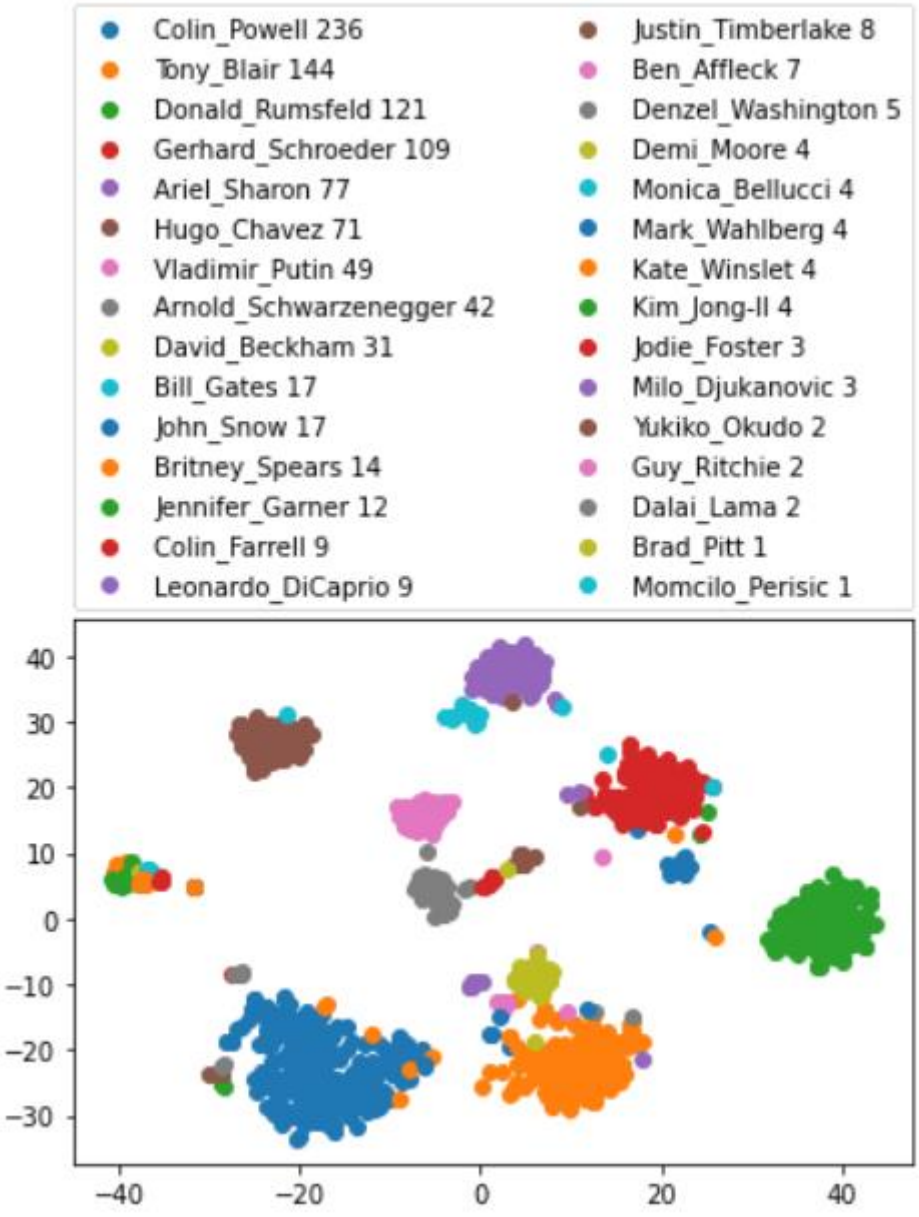
Gdje je m broj tripleta u trening setu, a notacija $[z]_+ = \max(z, 0)$. Dakle, da bi loss funkcija opadala, potrebno je da bude ispunjen uslov:

$$d(x^A, x^P) + \alpha \leq d(x^A, x^N)$$

U suprotnom, loss funkcija raste.

Tokom treniranja neuralne mreže, bitno je izabrati one triplete koje je teže razlikovati, tako da neuralna mreža može da nauči što više. To znači da je bitno izabrati one triplete za koje važi da je distance između anchor image i positive image slična distanci između anchor image i negative image. Biranjem ovakvih tripleta, tjeramo neuralnu mrežu da se potrudi što više. Jednom kada je mreža istrenirana na nekim podacima, može da se koristi za generisanje mjerenja za bilo koje slike, čak i one koje nikada prije nije ni vidjela.

Umjesto treniranja modela ispočetka, koje je i vremenski i računski zahtjevno, i zahtijeva veliki broj slika, iskorišten je već istreniran model mreže. S toga je samo potrebno učitati težine neuralne mreže u Kerasu pomoću load_weights(openface.h5).



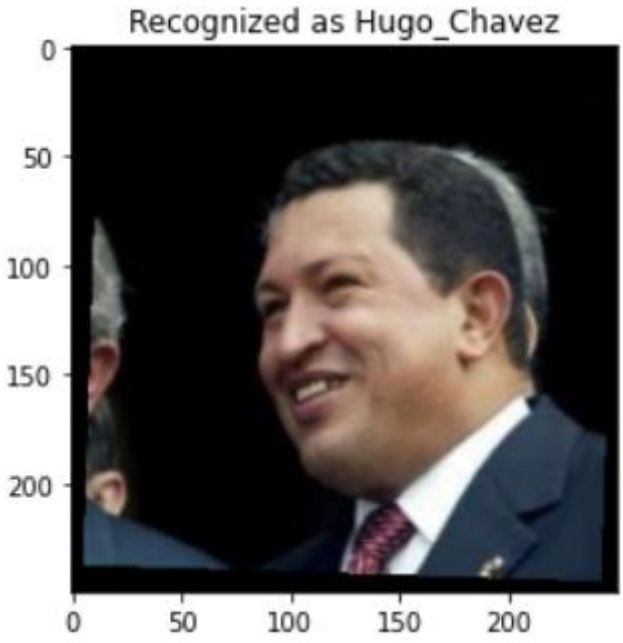
Slika 4.1: Vizualizacija višedimenzionih mjerenja u 2D prostoru.

5 Prepoznavanje lica

Nakon što je CNN dala obilježja slika u obliku vektora sa 128 elemenata, može se uraditi podjela podataka na trening i test skup. Trening podaci se koriste za obučavanje klasifikatora. U ovom radu, radi demonstracije iskorištena su 3 različita klasifikatora: K najbližih susjeda (KNN), metoda nosećih vektora (SVM) i neuralna mreža sa jednim skrivenim slojem, i sa 100 neurona u prvom (ulaznom) sloju i 50 neurona u skrivenom sloju.

Rezultati su sljedeći:

Preciznost na test skupu kada se koristi KNN klasifikator (k=7) iznosi 94%, slično je i kod SVM klasifikara, gdje preciznost na test skupu iznosi 95%, a prilikom obučavanja neuralne mreže za 300 epoha, na validacionom skupu je dostignuta preciznost od 95%.



Slika 5.1: Prepoznavanje lica.

6 LITERATURA

[1] https://www.kaggle.com/atulanandjha/lfwpeople
[2] https://hal.inria.fr/file/index/docid/548512/filename/hog_cvpr2005.pdf
[3] http://cmusatyalab.github.io/openface/models-and-accuracies/
[4] https://github.com/iwantooxxoox/Keras-OpenFace
[5] https://github.com/krasserm/face-recognition/blob/master/model.png