Primena metoda dubokog ucenja u sintezi sistemi za prepoznavanje lica

- Uvod
- Istorija
- Pregled sistema
- Teorijska osnova
- Detekcija lica
- Poravnanje lica
- Ekstrakcija vektora obelezja
- Pretraga vektora obelezja
- Implementacija u programskom jeziku Python
- Dodatna poboljsanja i problemi
- Zakljucak
- Literatura i reference

UVOD

Glavni problem koji razmatram u ovom radu jesu komponente koje cine jedan sistem za prepoznavanje lica. Jos od ranih dana sa pojavom kamera, rodila se potreba za sistemima koji bi mogli da identifikuju osobe na njima. Tokom vremena su se razne metode smenjivale, od prepoznavanja pokreta, otisaka prstiju, do prepoznavanja lica. U ovom radu ce biti reci upravo o tehnologijama koje se koriste za prepoznavanje lica, konrektno, fokusiracemo se na tehnike dubokog ucenja u sintezi sistema za prepoznavanje lica. Ovakvi sistemi su slozeni i predstavljaju sintezu raznih tehnologija i metoda kako bi uspesno radili. Sam razvoj ovih sistema je zahtevan, i zahteva tim ljudi koji su sposobni za resavanje kako matematickih, tako i racunarskih problema. Ovi problemi proizilaze iz potrebe za visokom tacnoscu sistema, kao i od varijacija usled koriscenja razne opreme i alata. Svakoga dana se sve vise softvera zasniva upravo na ovoj tehnologiji, a to podrazumeva njenu ispravnost, robusnost i tacnost. Jedna od tehnika koja je omogucila nagli razvoj ove oblasti jesu duboke neuronske mreze. Pored ovoga, ubrzan razvoj tehnologija i racunarskih komponenti omogucili su znatno brzi razvoj i treniranje ovakvih mreza, o cemu ce biti reci u nastavku ovog rada.

Glavni cilj mog istrazivanja u ovom radu jeste sinteza sistema za prepoznavanje lica koristeci vec postojece metode za detekciju, ekstrakciju vektora obelezja, i njihovo uporedjivanje radi dobijanja zelenih rezultata.

U svom radu koristicu analiticke metode kako bi svaku celinu razlozio na delove i bolje objasnio. Sam sistem za prepoznavanje lica je jedna slozena celina koja ukljucuje delove koji su se godinama razvijali i istrazivali. Ovo znaci da se kombinacijom ovih podsistema mogu dobiti novi sistemi sa specificnom namenom ili performansama. Ovi delovi u prokucionom sistemu ne mogu da rade jedan bez drugog, dok se je prilikom istraizivanja moguce preskociti neke od njih. Uzmimo za primer detekciju lica. Ukoliko je cilj sistema samo prepoznavanje, a za testiranje, razvoj i upotrebu se koriste slike koji sadrze samo detektovana lica, onda se sam korak detekcije moze preskociti. S obzirom na to da ovo cesto nije slucaj, fokusiracemo se na sintezu kompletog sistema.

U cilju uporedjivanja performansi sa javno dostupnim rezultatima koristicemo setove podataka koji su opsteprihvaceni u ovoj oblasti. Ovo nam omogucava realniju sliku o

performansama sistema. Dodatno, koristicemo podatke koji nisu cesto korisceni, ali su javno dostupni. Rec je o slikama koje na prvi pogled coveku deluju tesko za prepoznavanje. Kao treci nacin testiranja, bice koriscena kamera i snimak sa iste kako bi videli ponasanje sistema u realnom vremenu. Kod komercijalnih sistema je ovo kljucan korak gde moze doci do velikih gresaka. Uzmimo za primer dve osobe koje su jedna pored druge, setaju i okrecu se. Ukoliko se prepoznavanje ne radi u svakom frame-u, moze doci do zamena njihovoh identiteta prilikom pracenja ukoliko ovo nije odradjeno na pravi nacin.

1. Pregled sistema

Analiza lica predstavlja jedan od bitnih procesa u nasim zivotima. Ljudi analizom lica prikupljaju bitne podatke o drugim osobama. Ovo ukljucuje podatke o broju godina, polu, rasnoj pripadnosti. Takodje mozemo prepoznati da li je osoba srecna ili tuzna, ili pak neku drugu emociju. Pokreti usana su vazni u oblasti prepoznavanja govora, kao i sve popularnijoj oblasti kao sto je generisanje laznih snimaka. Metode analize lica nam mogu reci gde je usmeren pogled neke osobe, odnosno sta privlaci njenu paznju, i ovo moze biti posebno interesantno u marketingu i sopovima, kazinima. U medicini ove metode mogu biti od koristi za prepoznavanje nekih bolesti, poput autizma koji se odlikuje time sto osobe imaju poteskoca da iskazu svoje emocije. Sve navedene metode koriste kako ljudi, tako i racunari.

U ovom radu cemo se fokusirati na metode koje se koriste u procesu detekcije lica, njegove ekstrakcije, obrade i zatim prepoznavanja.

Na pocetku ovog rada je prvo bitno da uvrdimo sta je zapravo prepoznavanje lica. Svaka osoba ima karakteristicno lice, i to je ono sto nas cini unikatnima. Vecina metoda se bazira upravo na ovoj cinjenici, i njihov cilj je ekstrakcija ovih obelezja (features) za svaku osobu, a zatim i njihova klasifikacija na osnovu odredjenih parametara.

Treba razlikovati algoritme za prepoznavanje po vise kriterijuma. Osnovna klasifikacija je na algoritme zasnovane na geometri (Geometry based) i algoritme zasnovane na sablonima (template based). Geometrijski zasnovani algoritmi analiziraju odredjena podrucja i geometrijske veze na njima. Zbog ovoga su i poznati kao algoritmi zasnovani na obelezjima (feature). Sa druge strane su algoritmi zasnovani na sablonima, i u ovu grupu spadaju: metoda nosecih vektora (SVM),

analiza glavnih komponenti (PCA), linearna diskriminantna analiza (LDA), kernel metode i jos mnogo drugih.

Dodati jos u uvodu…
Dodati osnovne stvari o koriscenim metodama u narednom delu.
Konvolucija/ Graf konvolucija
Rigidne transformacije
Pooling slojevi
Loss funkcije
Augmentacija podataka
MobileNet

RetinaFace

Prvi korak u procesu prepoznavanja lica je njegova detekcija. Osnovna ideja procesa detekcija je pronalazenje svih lica na slici, odnosno njihovih koordinata. Ove koordinate sluze za ekstrakciju lica sa slike, te se u nastavku radi samo sa slikama koje sadrze lice. Takodje se vrlo cesto koordinate cuvaju ukoliko je rec o radu sa video snimcima radi iscrtavanja. Pored ovog, cuvanje originalne slike i koordinata (bounding box-a ili bbox-a) imamo mogucnost ponovne ekstrakcije lica sa raznim scale faktorima. Sto moze biti korisno kao ulaz u neke druge neuronske mreze. Primer ovoga su anti-spoofing mreze koje imaju za cilj predikciju da li osoba na slici ili snimku prava (real) ili lazna (fake).

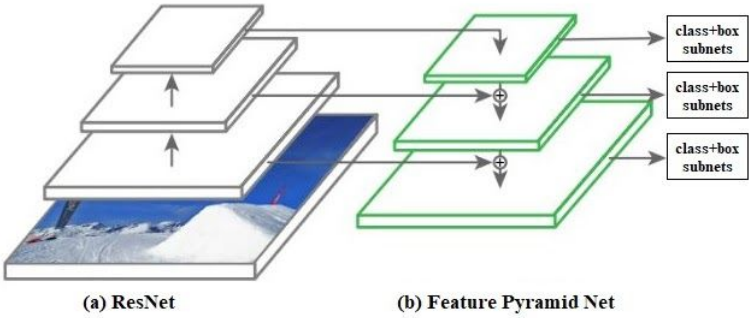
Postoji zaista veliki broj razvijenih ideja i projekata na temu detekcije, kako objekata, tako i lica. Jedna implementacija se izdvaja od drugih, rec je o RetinaNet mrezi. Ovde je rec o RetinaNet mrezi, dok ce u nastavku biti reco o RetinaFace-u. Razlika je u tome sto je RetinaNet prva imlementacija koja je namenjena generalno detekciji objekata.

Osnovna ideja sa Retina detektorom je bila u uraditi sve predickije u jednoj fazi. U 2019. godini, state-of-the-art (SOTA) detektori su bili bazirani na mehanizmu sa dva faze (takozvani two-stage). Prva faza je podrazumevala generisanje skupa kandidata za lokaciju na kojoj je moguce naci trazeni objekat, dok se u druga faza sastoji iz klasifikacije svakog kandidata u jednu od klasa.

RetinaNet je jednofazni (single stage) detektor koji se sastoji jedne mreze (bacbone) i dve mreze sa specificnom funkcijom. Kao backbone se moze koristiti bilo koja od poznathih arhitektura (VGG, MobileNet, ResNet) i osnovni cilj ove mreze je racunanje konvolucione feature mape. Nakon toga, prva podmreza radi klasifikaciju na osnovu izlaza backbone mreze, dok druga podmreza proracunava bounding box regresiju.

Pored toga sto je ovaj detektor bio single stage, podrzavao je koncept piramida. RetinaNet je zasnovan na konceptu feature piramida (Feature Pyramid Network - FPN). Ovo je omoguceno koriscenjem FPN mreze kao backbone mreze. Osnovni princip rada je da FPN

mreza radi augmentaciju standardne konvolucione mreze sa vrha ka dnu sa lateralnim konekcijama (bocne veze). Ovo omogucava mrezi da efikasno konstruise piramidu na sa razlicitim scale faktorima iz jedne slike. Svaki level sa piramide se moze koristiti za detektovanje objekata u razlitoj razmeri.



[https://arxiv.org/pdf/1708.02002.pdf]

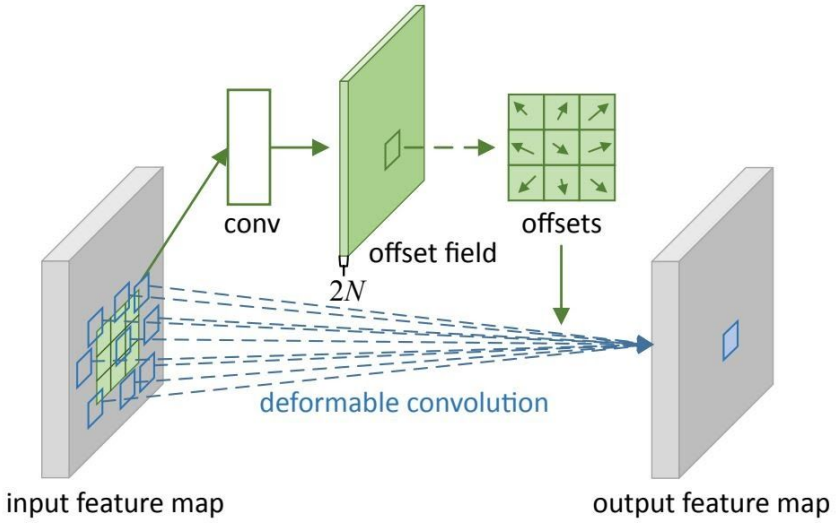
Jedan od velikih problema koji se pojavljivao je bila detekcija lica razlitih velicina u nekotrolisanim uslovima (prirodi, guzvama u gradu). Posto vec poznati koncept piramida moze restiti probleme detektovanja sa razlicitim scale faktora, sledeca mreza koristi iste principe. RetinaFace je mreza nesto novijeg porekla i zasnovna je na istim principima na kojima se zasniva i RetinaNet, ali je namenjena iskljucio detekciji lica.

RetinaFace predstvalja single stage detektor lica. Tvorci ovog modela su uneli nove ili unapredilivec postojece metode koriscene u ovu svrhu, kao sto su multi-task obucavanje za istovremenu predikciju sigurnosti, bounding box-a, 5 kljucnih tacaka na licu, i 3D poziciju (u originalnoj implementaciji).

Ono sto je novo je 5 kljucnih tacaka (keypoint-a) koji ce se kasnije koristiti za poravnanje lica.

Kako bi se poboljasla detekcija lica, ili takozvanih Hard detekcija, koriscen je koncept modelovanja konteksta. Takozvana Hard lica su teska za detekciju zbog nedostatka vizualne konzistentosti, pozicije ili konteksta [https://arxiv.org/pdf/1803.07737.pdf]. Osnovna ideja je da se mreza moze da nauci ne samo feature koji su karakteristicni za lice, vec i kontekstualni deo kao sto je vrat ili telo.

Kako bi se povecali efekti modelovanja nelinearnih (ne-rigidnih) transformacija (scaling, shearing) korisca je mreza pod nazivom Deformable Convolution Network (DCN). Geometrijske varijace predstavljaju jedan od velikih problema u oblasti detekecije i prepoznavanja. Metoda koja se pokazala korisnom u prevazilazenju ovih problema je koriscenje deformabilne (eng. deformable) konvolucije.



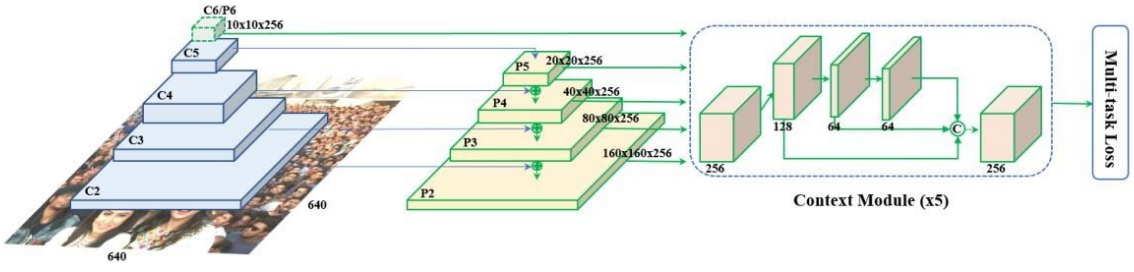
[https://arxiv.org/pdf/1703.06211.pdf]

Matemaika o deformable conv vs obicna conv.
i DCN v2 [https://arxiv.org/pdf/1811.11168.pdf]

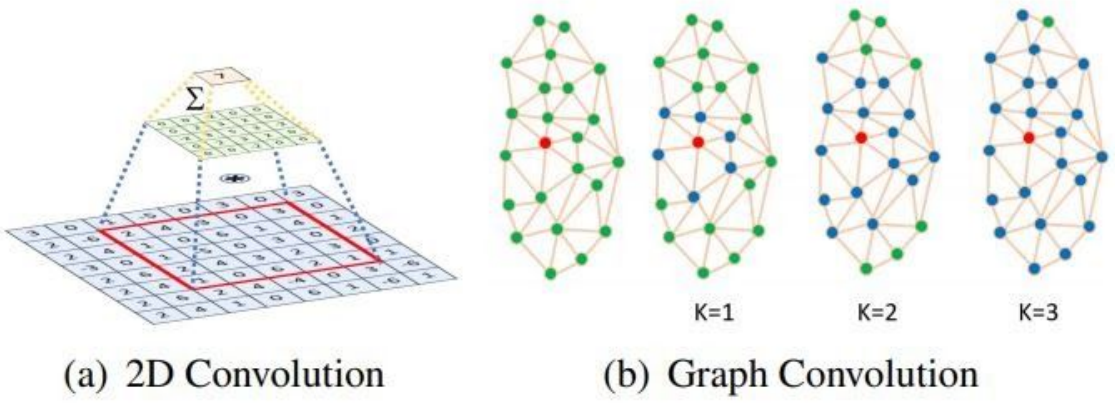
Kao sto je vec pomenuto, RetinaFace mreza je zasnovana na pricipu multi-tast obucavanja. Samim tim se namece koriscenje drugacije funkcije gubitka (loss funkcije). Loss funkcija koriscena u ovom slucaja je multi-tast loss [https://arxiv.org/pdf/1905.00641.pdf]:

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel}.$$

Ova funkcija se sastoji iz vise delova, gde prvi deo Lcls predstavlja sfotmax loss binarne klasifikacije (ima lica/nema lica), drugi deo je loss regresije bounding box-ova, zatim sledi regresioni loss za predikciju 5 kljucnih tacaka i dense regresioni loss.



Kako bi ubrzao proces detekcije koriscen je rakozvani mesh dekoder (mesh konvolucija i up-sampling). Ovo predstavlja vid graf konvolucionog metoda o kome je ranije bilo reci.



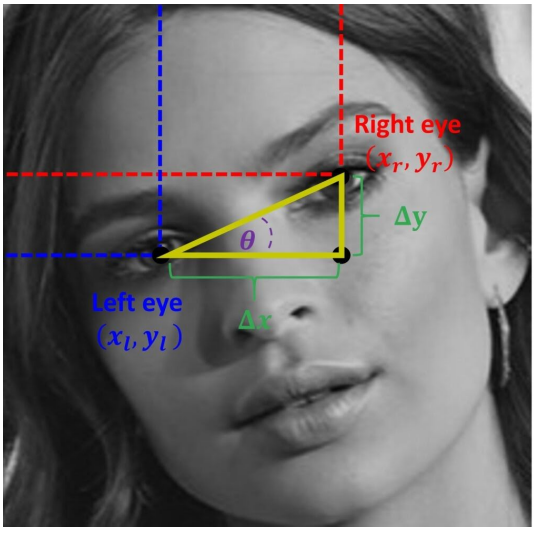
(a) 2D Convolution (b) Graph Convolution

Dodati jos o implementaciji ovde…

Poravnanje lica

Nakon procesa detekcije i ekstrakcije lica i 5 kljucnih tacaka, kako bi proces prepoznavanja bio sto uspesniji, potrebno je uraditi poravnanje lica.

Jedan od jednostavnih ali uspesnih metoda za ovo je pronalazenje arcus tangensa izmedju dva oka (odnosno ugla izmedja dva oka). Nakon toga, potrebno je izracunati rotacionu matricu.



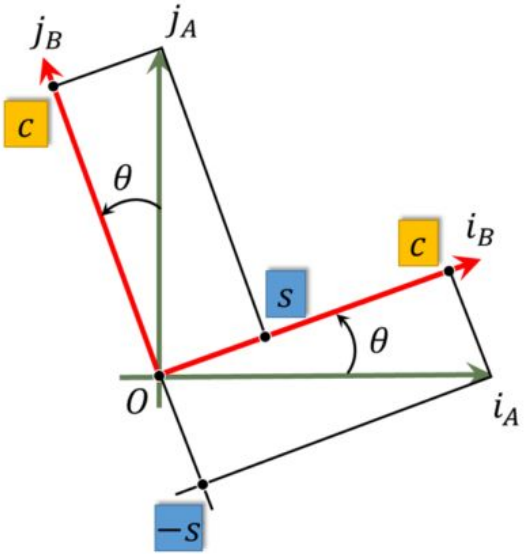
$$\Delta x = x_r - x_l$$

$$\Delta y = y_r - y_l$$

$$\theta = \arctan\frac{\Delta y}{\Delta x}$$

Ovde je bitno napomenuti da arctan funkcija u Numpy paketu vraca ugao u radijanima. Za dalju upotrebu je potrebno pretvoriti ga u stepene. Za ovo je samo potrebno pomnoziti sa 10 i podeliti sa PI.

Nakon ovoga, potrebno je izracunati rotacionu matricu.



$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

$$R_Z(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

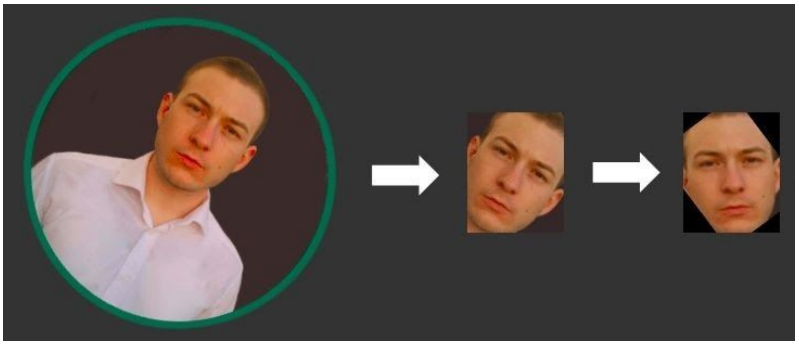
$$R_Y(k) = \begin{bmatrix} cos(k) & 0 & -sin(k) \\ 0 & 1 & 0 \\ sin(k) & 0 & cos(k) \end{bmatrix}$$

Sledeci korak je upotreba afinih transformacija kako bi postigli zeljeni efekat. Ovakva vrsta transformacija, odnosno preslikavanja preslikva tacke u tacke, prave u prave, ravni u ravni. Kod ovakvih transformacija, par paralelnih pravi ostaje paralelan i nakon preslikavanja, ali uglovi izmedju pravih ili razdaljine izmedju tacaka ne moraju nuzno da ostanu isti.

$${}^{B}\mathrm{P} = {}^{B}_{A}\mathrm{R}\ {}^{A}\mathrm{P}$$

$$\begin{bmatrix} {}^{B}\mathrm{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}\mathrm{R} & \mathbf{0} \\ \mathbf{0}^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{A}\mathrm{P} \\ 1 \end{bmatrix}$$

Ceo proces detekcije i poravnanja lica je prikazan na sledecoj slici.



Ekstrakcija vektora obelezja

Nakon sto imamo sliku poravnanog lica, sledeci korak u procesu prepoznavanja je ekstrakcija vektora obelezja (feature vektora). Tokom godina su se smenjivali razne metode sa istom namenom, ali sa razlictim metodama ekstrakcije vektora, kao i velicine vektora. Ranije SOTA implementacije poput FaceNet mreze su bile zasnovane na 128 dimenenzionalnim vektorima. Novije metode imaju mogucnost ekstrakcije 512-dimenzionalnih obelezja, pa cak i 1024. Kao optimalan broj feature-a se pokazao 512.

Dodati poredjenje…..

Metode poput FaceNet [https://arxiv.org/pdf/1503.03832.pdf] mreze su za cilj imale direkto ucenje vektora obelezja zasnovane na triplet loss funkciji. Ideja je minimiziranje distance izmedju vektori iste osobe (positive), dok se vektori druge osobe (negative) udaljavalju. Ovo je podrazumevalo da u svakom trenutku tokom treninga imamo tri vektora (ancor, positive, negative). Proces odabira tripleta je jako zahtevan i spor, sto je bio prvi nedostatak ovakvog i slicnih metoda.

Jedan od implementacija koja je postigla SOTA rezultate je ArcFace [https://arxiv.org/pdf/1801.07698.pdf]. Osnovna razlika i ideja je bila napraviti klasifikator koji moze da razdvoji razlicite identitete u trening setu na osnovu odredjene loss funkcije, kao i koriscenje 512 dimenzionalnih vektora obelezja. Problem sa triplet loss obucavanjem je i eksponencijalni skok broja kombinacija kod velikih setova podataka, dok je problem sa tradicionalnim funkcijama poput softmax je sto se linearna transformaciona matrica povecava linearno, sto nije problem kod manjih setova podataka, ali kod velikih setova ili produkcionih sistema neupotrebljivo.

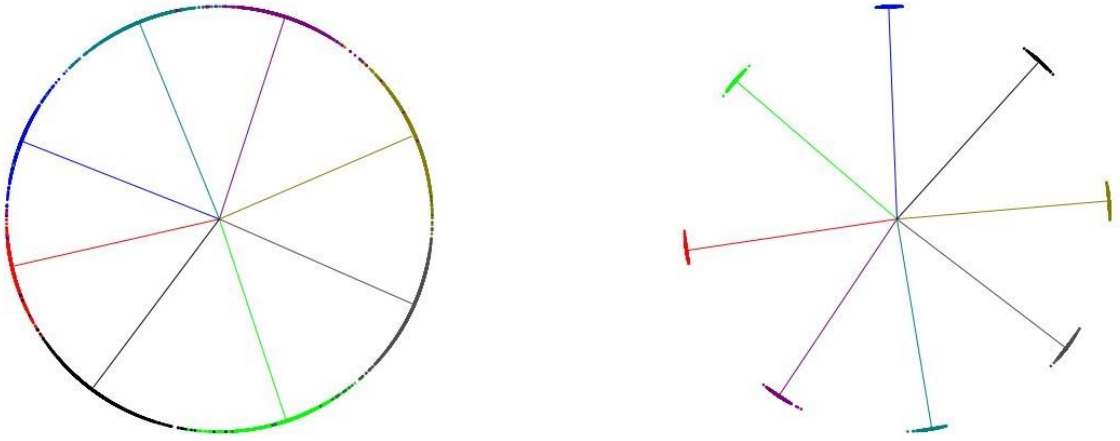
Kako bi se povecala margina izmedju klasa, predstavljena je nova loss funkcija. Rec je o ArcFace loss funkciji.

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{i=1}^{n} e^{W_j^T x_i + b_j}}$$

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}$$

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}.$$

Na sledecoj slici je prikazano kako izgleda separacija 8 klasa koriscenjem Softmax i ArcFace funkcjie.



(a) Softmax (b) ArcFace

Kao backbone mreza je u originalnom radu koriscen ResNet, ali u nasem slucaju je izabrana lightweight arhitektura pod nazivom MobileNet.

Pretraga vektora obelezja

Ranije imeplemetnacije slicnih sistema [staviti reference] su koristile razne metode pretrage. Od najjednostavnihij poput brute force pretrage, k najblizih suseda (KNN), metode nosecih vektora (SVM), pa i neuronskih mreza za pretragu vektora i predickiju identitieta.

Pomenute metode su se pokazale kao spore, ili nedovoljno precizne. Stoga, bilo je potrebe za novim i brzim metodama pretrage velikih skupova podataka. Metoda koja se izdvaja je pretraga aproksimiranih k najblizih suseda koriscenjem Hierarchical Navigable Small World grafova (HNSW) [https://arxiv.org/ftp/arxiv/papers/1603/1603.09320.pdf].

Opisati KNN i opisati ANN
Opisati izmene koriscenjem HNSW

Implementacija u programskom jeziku Python

Dodatna poboljsanja

Zakljucak

Literatura i reference