

Detekcija i prepoznavanje lica koristeći duboko učenje i SQL bazu za pohranu

Antolić, Katarina

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:961552>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-02**



Repository / Repozitorij:

[University North Digital Repository](#)





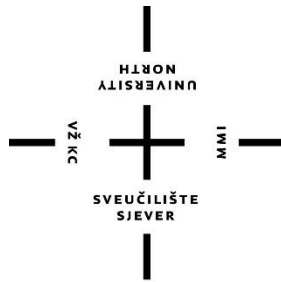
**Sveučilište
Sjever**

Diplomski rad br. 140-MMD-2024

**DETEKCIJA I PREPOZNAVANJE LICA KORISTEĆI
DUBOKO UČENJE I SQL BAZU ZA POHRANU**

Katarina Antolić, 0016139654

Varaždin, rujan 2024. godine



Sveučilište Sjever

Multimedija

Diplomski rad br. 140-MMD-2024

DETEKCIJA I PREPOZNAVANJE LICA KORISTEĆI DUBOKO UČENJE I SQL BAZU ZA POHRANU

Student

Katarina Antolić, 0016139654

Mentor

izv.prof.dr.sc. Emil Dumić

Varaždin, rujan 2024. godine

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ diplomski sveučilišni studij Multimedija

PRISTUPNIK Antolić Katarina

IMBAG

0016139654

DATUM 26.08.2024.

KOLEGIJ

Računalni vid

NASLOV RADA Detekcija i prepoznavanje lica koristeći duboko učenje i SQL bazu za pohranu

NASLOV RADA NA ENGL. JEZIKU Face detection and recognition using deep learning and SQL storage database

MENTOR Emil Dumić

ZVANJE

izv.prof.dr.sc.

ČLANOVI POVJERENSTVA

- doc. art. dr. sc. Mario Periša - predsjednik
- izv. prof. art. dr. sc. Robert Geček - član
- izv. prof. dr. sc. Emil Dumić - mentor
- doc. dr. sc. Andrija Bernik - zamjenski član
-

Zadatak diplomskog rada

BRD: 140-MMD-2024

OPIS

U ovom radu bit će napravljen sustav za detekciju i prepoznavanje lica, implementiran u web sučelje koje pohranjuje značajke lica u sql bazu, a nova lica će se prepoznati usporedbom sa stvorenom bazom.

Prepoznavanje lica (FR) je istaknuta biometrijska tehnika za identifikaciju i provjeru autentičnosti te se koristi u mnogim područjima, kao što su vojska, financije, javna sigurnost i općenito u svakodnevnom životu. Algoritmi za prepoznavanje lica mogu se sastojati se od pretkorska (detekcija lica i eventualno poravnanja lica), dok se samo prepoznavanje djeli na provjeru (1:1 problem) i identifikaciju (1:N problem). Opisat će se mjere učinkovitosti kod prepoznavanja lica: TMR@FMR (TAR@FAR, TPR@FPR) ili FNMR@FMR (FNR@FPR) za neki prag FMR-a (za provjeru); TPIR@FPIR ili FNIR@FPIR za neki prag FPIR-a (za identifikaciju, otvorena baza); te Rang-N (za identifikaciju, zatvorena baza). Opisat će se metode dubokog učenja za detekciju objekata (i time lica) poput Mobilenet (v1, v2, v3), ResNet, YOLO grupe neuronskih mreža, te MTCNN koja se može koristiti za detekciju i poravnanje lica. Opisat će se algoritmi za prepoznavanje lica koristeći duboke neuronske mreže poput FaceNet (i OpenFace), DeepFace, DeepID i VGGFace algoritama. Opisat će se noviji radovi vezani za odabir funkcije gubitka: AdaFace, MagFace i ArcFace. Također će biti opisane različite prijetnje prepoznavanju identiteta na temelju biometrije lica.

Praktični dio rada bit će stvaranje web sučelja koje pohranjuje značajke detektiranog lica u sql bazu koristeći neke od opisanih metoda dubokog učenja. Ispitat će se različiti algoritmi za detekciju i prepoznavanje lica, te će se usporediti različiti algoritmi za prepoznavanje lica također koristeći različite metode udaljenosti vektora značajki (npr. kosinusna ili euklidska udaljenost). Vrijeme izvođenja će se također uzeti u obzir. Algoritmi za prepoznavanje će se usporediti koristeći ranije opisane mjere učinkovitosti. Bit će korištena neka od postojećih baza lica (npr. Labeled Faces in the Wild, LFW) ili će biti napravljena vlastita baza.

ZADATAK URUČEN 02.09.2024.

POTPIS MENTORA

Emil Dumić

Sažetak

Cilj ovog rada je spojiti moje područje interesa, baze podataka, s raznim drugim područjima kao što su područje biometrije i detekcije lica, područje izrade web aplikacija te sve zajedno povezati u kompaktnu cjelinu. U teorijskom dijelu rada objašnjen je i prikazan sam pojam biometrije, detekcije i identifikacije lica, prikazani su razni modeli dubokog učenja za detekciju lica. Objašnjeni su i algoritmi koji se koriste, a isto tako i razne poteškoće i izazovu u procesu obrade slika u svrhu identificiranja pojedinaca. Praktični dio rada odnosi se na testiranje opisanih algoritama i modela učenja, izradu web aplikacije i korisničkog sučelja za testiranje algoritama te na izradu baze podataka koja služi kao poveznica između tih komponenti. U praktičnom radu korištene su razne tehnologije i alati primjereni ovoj temi, a u teorijskom dijelu korišteni su relevantni izvori literature poput znanstvenih članaka, knjiga, tehnoloških blogova i sve ostale provjerene literature.

Ključne riječi: biometrija, detekcija lica, identifikacija, duboko učenje, modeli, algoritmi

Popis korištenih kratica

PARAFAE	Passport verification and border control system Francuski sustav za graničnu kontrolu.
ASM	Active shape model Generativni model poravnavanja lica.
AAM	Active appearance model Generativni model poravnanja lica.
PDM	Point distribution model. Generativni model poravnanja lica.
CLM	Constrained local model. Generativni model poravnanja lica.
SIC	Simultaneous inverse compositional algorithm Algoritam generativnih metoda za poravnanje lica.
POIC	Project-out inverse compositional algorithm Algoritam generativnih metoda za poravnanje lica.
NLP	Natural Language Processing Područje računalne znanosti za procesiranje ljudskog govora u računalni.
CVPR	Conference on Computer Vision and Pattern Recognition Godišnja konferencija o računalnom vidu.
CmBN	Cross mini-batch normalization Normalizacija koja radi na manjim serijama podataka.
ELAN	Extended efficient layer aggregation network Mreža za treniranje YOLO modela.
PGI	Programmable gradient information Novi koncept YOLOv9 arhitekture.
IoT	Internet of things Uređaji koji se u svom radu spajaju na Internet.
TRR	True Rejection Rate Omjer istinitog odbacivanja (istinito negativan omjer).
FAR	False Acceptance Rate Omjer lažnog prihvaćanja (lažno pozitivan omjer).
FRR	False Rejection Rate Omjer lažnog odbacivanja (lažno negativan omjer).

EER	Equal Error Rate Omjer jednake pogreške.
ROC	Receiver Operating Characteristic Curve Operativna karakteristika prijavnika.
GAR	Genuine Acceptance Rate Omjer istinitog prihvaćanja (istinito pozitivni omjer).

Sadržaj

1.	Uvod	1
2.	Detekcija i prepoznavanje lica	2
2.1.	Biometrija i biometrika	3
2.1.1.	<i>Povijesni pregled biometrike.....</i>	<i>3</i>
2.2.	Primjene prepoznavanja lica u različitim domenama.....	5
2.3.	Proces poravnanja i detekcije lica	7
2.4.	Prijetnje i izazovi u detekciji i identifikaciji lica.....	9
3.	Duboko učenje.....	12
3.1.	CNN	14
3.1.1.	<i>Arhitektura CNN-a</i>	<i>14</i>
3.1.2.	<i>Proces treniranja.....</i>	<i>15</i>
3.1.3.	<i>Detekcija objekata.....</i>	<i>16</i>
3.1.4.	<i>Izazovi i ograničenja.....</i>	<i>16</i>
4.	Metode dubokog učenja za detekcija objekata na slici	17
4.1.	YOLO metoda	17
4.1.1.	<i>YOLOv1.....</i>	<i>19</i>
4.1.2.	<i>YOLOv2.....</i>	<i>20</i>
4.1.3.	<i>YOLOv3.....</i>	<i>21</i>
4.1.4.	<i>YOLOv4.....</i>	<i>22</i>
4.1.5.	<i>YOLOv5.....</i>	<i>23</i>
4.1.6.	<i>YOLOv6.....</i>	<i>23</i>
4.1.7.	<i>YOLOv7.....</i>	<i>24</i>
4.1.8.	<i>YOLOv8.....</i>	<i>24</i>
4.1.9.	<i>YOLOv9.....</i>	<i>24</i>
4.2.	MobileNet.....	26
4.2.1.	<i>MobileNetV2</i>	<i>29</i>
4.2.2.	<i>MobileNetV3</i>	<i>30</i>
4.3.	MTCNN	31
5.	Metode dubokog učenja za klasifikaciju slika.....	32
5.1.1.	<i>ResNet</i>	<i>32</i>
5.1.2.	<i>VGG</i>	<i>34</i>
6.	Algoritmi za prepoznavanje lica.....	36
6.1.	FaceNet.....	36
6.2.	DeepFace	38
6.3.	DeepID i M ² Deep-ID	40
6.4.	VGGFace i VGGFace2	42
7.	Funkcija gubitka u prepoznavanju lica	44
7.1.	AdaFace.....	45
7.2.	ArcFace	47
7.3.	MagFace	48

8.	Mjere učinkovitosti prilikom detekcije lica	50
9.	Praktični rad.....	53
9.1.	Opis ideje.....	53
9.2.	Korištene tehnologije	53
9.3.	Arhitektura sustava.....	54
9.3.1.	<i>Preduvjeti za pokretanje i korištenje aplikacije</i>	<i>54</i>
9.3.2.	<i>SQL Server</i>	<i>64</i>
9.3.3.	<i>BitBucket</i>	<i>65</i>
9.4.	Hodogram rada aplikacije i opis skripti	66
9.4.1.	<i>Opis skripti</i>	<i>68</i>
9.5.	ResNet i FaceNet usporedba	70
9.6.	Testiranje mjera učinkovitosti	70
9.6.1.	<i>FaceNet</i>	<i>71</i>
9.6.2.	<i>ResNet</i>	<i>74</i>
10.	Zaključak	78
11.	Popis slika	79
12.	Literatura.....	82

1. Uvod

U doba ubrzanog tehnološkog napretka, koncepti poput umjetne inteligencije, strojnog učenja i dubokog učenja postaju sveprisutni u našem društvu i potiču brojne rasprave i interes javnosti. Unatoč njihovoj popularnosti, često se primjećuje nedostatak detaljnijeg razumijevanja ovih pojmova među širom publikom. Umjetna inteligencija, prema definiciji Johna McCarthyja, predstavlja znanstveno i inženjersko područje koje se bavi stvaranjem inteligentnih strojeva, odnosno računalnih programa sposobnih donošenja odluka. Iako se često povezuje s ljudskom inteligencijom, granice umjetne inteligencije ostaju definirane biološkim metodama koje su razumljive čovjeku. Duboko učenje, ključan dio umjetne inteligencije, i strojno učenje, često promatrani kao dio iste cjeline, imaju značajnu ulogu u razvoju suvremenih tehnoloških rješenja. Primarna svrha strojnog učenja je razvijanje algoritama koji mogu donijeti predviđenu odluku ili provesti specifičnu radnju na temelju tih podataka. Iako su algoritmi ono nešto što sve pokreće, tema ovog rada nije vezana samo uz algoritme, već i uz metode koje se njima koriste.

Pojam biometrije svima je poznat, a činjenicu da svoje pametne uređaje otključavamo slikom svog lica, uzimamo zdravo za gotovo. Rijetko tko promišlja o tome što se odvija u pozadini kada nas neki algoritam prepozna na slici na kojoj smo primjerice označeni na Instagramu, ili što se događa u pozadini kada se prijavimo u neki sustav uz pomoć detekcije našeg lica. Upravo u ovom radu bit će prikazane neke od najpoznatijih metoda za kojima poseže duboko učenje prilikom detekcije ljudi, ali i objekata na raznim slikama i videozapisima. Na kraju svega, bit će prikazan i praktični primjer web aplikacije koja koristi neku od opisanih metoda i razne algoritme za detekciju i prepoznavanje lica. Usporedit će se rezultati koje daju pojedini algoritmi i modeli. Na kraju svega izvesti će se zaključak o cjelokupnoj temi s osvrtom na teorijski i praktični prikaz teme.

2. Detekcija i prepoznavanje lica

Ljudsko lice je ono nešto što svakog pojedinca čini jedinstvenim i posebnim. Građeno od velikog broja stanica, pora, različitih dimenzija, oblika i boje, ljudsko je lice nešto što oduvijek privlači pažnju. Kada upoznajemo nove osobe, uvijek prvo primijetimo lice, a i kada povezujemo osobu čijeg imena se ne možemo sjetiti, uvijek povezujemo lice s osobom. S obzirom kakvu važnu ulogu igra u identitetu svakoga od nas ponaosob, nije ni čudno da se prepoznavanjem odnosno detekcijom lica ljudi bave od prethistorijskog vremena. Sam pojam prepoznavanja lica ima mnogo različitih definicija od kojih je teško izvesti jednu univerzalnu. No, tehniku prepoznavanja lica najlakše se može definirati kao biometrijsku tehniku koja povezuje lice pojedinca u svrhu njegovog identificiranja [1].

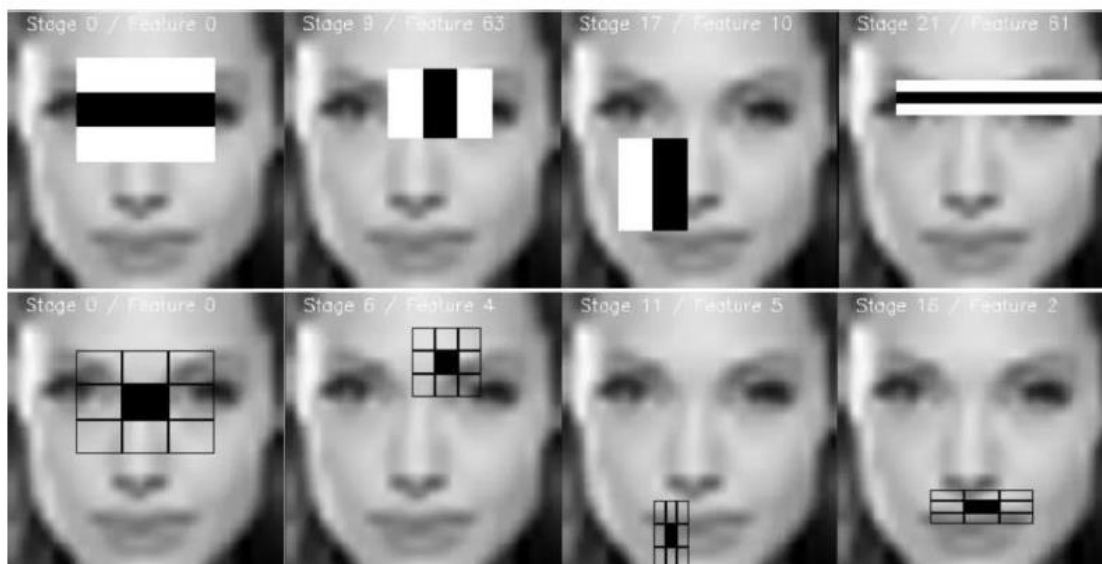
No, prethodna definicija povlači pitanje što su biometrijske tehnike i kako su povezane s detekcijom lica, a sve je detaljnije objašnjeno u nastavku teksta. Za što bolje razumijevanje tehnologije prepoznavanja lica, prvo je potrebno objasniti nekoliko općenitih pojmova poput biometrije, biometrijskih tehnika i slično. Biometrija je znanstvena disciplina koja predstavlja niz faktora za prepoznavanje ljudi na temelju njihovih fizičkih karakteristika. Biometrijskih tehnologija za prepoznavanje osoba ima mnogo, a najpoznatije su svakako prepoznavanje lica, otisak prsta, otisak dlana, prepoznavanje rožnice ili šarenice i slično [2].

U ovom dijelu rada naglasak je na biometrijskoj tehnici prepoznavanja i identifikacije osoba pomoću detekcije i prepoznavanja lica. Sa sveopćim napretkom tehnologije razvija se i tehnologija detekcije i prepoznavanja lica. Prema Adjabiju i ostalima, počeci tehnologije prepoznavanja lica sežu sve do 1960-ih godina kada su započeli prvi pokušaji automatizacije procesa prepoznavanja lica na slikama. Te su se rane metode oslanjale na ručno definiranje karakteristika lica poput očiju, nosa i usta, što je ograničilo njihovu točnost i učinkovitost. Međutim, napredak u domeni tehnologije odnosno računala i pojavom interneta i mogućnosti prikupljanja velikog broja podataka, započeo je i razvitak algoritama za detekciju i prepoznavanje lica [3].

Tijekom 1990-ih, Viola i Jones predstavili su Viola-Jones algoritam, koji je značajno poboljšao točnost detekcije lica kao početni korak u prepoznavanju lica. Svoje prvo korištenje algoritam je doživio 2001. godine, a temeljio se na korištenju Haara i AdaBoost značajki. Haar značajke su jednostavne pravokutne značajke koje se izračunavaju oduzimanjem zbroja intenziteta piksela u bijelom dijelu slike od zbroja intenziteta piksela u crnom području slike. AdaBoost algoritam koristi se za odabir najboljih značajki iz danog skupa podataka. Jedna od najvažnijih prednosti Viola-Jones algoritma je njegova brzina upravo zbog korištenja AdaBoost značajki koje smanjuju vrijeme potrebno za izračun. Na slici 2.1 u nastavku prikazan je

pojednostavljeni način rada ovog algoritma, odnosno njegova detekcija objekata uz pomoć oduzimanja spomenutih intenziteta piksela sa bijelog i tamnog dijela slike [4].

Viola-Jones algoritam poslužio je kao temelj za kasnije tehnike i odigrao ključnu ulogu u širokom prihvaćanju tehnologije prepoznavanja lica.



Slika 2-1 Način rada Viola-Jones algoritma za detekciju lica (Preuzeto 5.2.2024. s https://medium.com/@Andrew_D/computer-vision-viola-jones-object-detection-d2a609527b7c)

2.1. Biometrija i biometrika

Biometrija je sinonim za biološku statistiku, a važno je poznavati razlike između dvaju sličnih pojmova, biometrije i biometrike. Biometrija se može definirati kao automatizirao prepoznavanje ljudi na temelju njihovih fizičkih obilježja i ponašanja. Biometrika s druge strane predstavlja metode u matematici i statistici koje pomažu prilikom analiziranja bioloških podataka koji se koriste u biometriji prilikom identificiranja osoba. U ovom je dijelu fokus na povijesnom razvoju i korištenju biometrike sve od ranih početaka do modernog korištenja ove discipline u biometrijske svrhe prepoznavanja i identifikacije ljudi.

2.1.1. Povijesni pregled biometrike

Još prije otprilike 30 000 godina, praljudi su koristili primitivne načine izražavanja i predstavljanja sebe kao pojedinaca, pa se tako u mnogim prapovijesnim špiljama nalaze crteži s otiscima nečijeg dlana što se povezuje s prvim pokušajima identificiranja. Pola tisućljeća prije

nove ere, Babilonci su koristili otiske prstiju kao potpis na glinenim pločama koje su korištene u nekim poslovnim transakcijama onog doba. U drevnoj Kini roditelji su koristili otiske prstiju i stopala svoje djece kako bi ih razlikovali, a stari Egipćani su na temelju fizičkog opisa trgovaca razlikovali nove od starih trgovaca, točnije, na taj su način identificirali pojedince prilikom odlaska na trgovinske rasprave. U novijoj svjetskoj povijesti, prekretnicu čini rad nizozemskog znanstvenika Govarda Bidloo's koji u svojoj knjizi "Anatomy of the Human Body" prvi put detaljno opisuje dijelove ljudske kože i kako ti dijelovi omogućuju identifikaciju pojedinca. Godinu dana kasnije, nastavno na Bidloo's-ov rad, talijanski sveučilišni profesor Marcello Malpighi dao je svoj doprinos razvitku biometrike detaljno objasnivši grebene, spirale i petlje koje se nalaze na ljudskim prstima, a koje su prvi korak prilikom pregledavanja otiska prstiju kako bi se zaključilo o kome je riječ. S razvitkom prometne infrastrukture i razvojem industrije, ljudi su postali skloniji putovanjima i migracijom diljem svijeta. Upravo iz tog razloga javila se potreba identificiranja i prepoznavanja pojedinaca. Jedan od najboljih primjera iz 19. stoljeća je razvoj Bertillonovog sustava raspoznavanja ljudi. Ovaj sustav zasnivao se na mjerenju tjelesnih dimenzija bilježenih na papirima koji su se kasnije razvrstavali prema visini, duljini, debljini ili nekom drugom obilježju pojedinog dijela tijela. No, s obzirom na to da postoji velik broj ljudi koji dijele ista fizička obilježja u pogledu tjelesnih dimenzija, ovaj je sustav ubrzo odbačen. Ipak, zahvaljujući ovom sustavu, kasnije se razvila cijela nova disciplina nazvana antropometrija. Godine 1892. prvi put se detaljnije razvija proces identifikacije prema otisku prstiju. Engleski znanstvenik Francis Galton tada je detaljno predstavio karakteristike otiska i njihove razlike na svih desetak prstiju, a te se metrike nazivaju Galtonovi detalji i koriste se i dan danas. Nastavno na njegovo istraživanje, 1986. godine inspektor bengalske policije Edward Henry uspostavio je prvu bazu otisaka prstiju korištenu u londonskoj policiji. Uzor u njegovoj klasifikaciji otisaka prstiju uskoro je pronašao FBI, a također još mnoge policijske postaje tog doba. Nekoliko godina kasnije događa se jedna od značajnijih prekretnica kada 60-ih godina prošlog stoljeća s razvojem započinje prvi poluautomatski sustav za detekciju lica kojeg je na zahtjev američke vlade razvio Woodrow Bledsoe. Poluautomatska detekcija zasnivala se na definiranju ušiju, očiju, usta i nosa na fotografijama prema kojima se kasnije moglo izvući zaključke o omjeru i udaljenosti spomenutih značajki kod svake osobe pojedinačno. Krajem tog desetljeća, 1969. godine američka obavještajna služba FBI automatizirala je i proces prikupljanja i obrađivanja otiska prstiju. Otprilike u isto doba počelo se raditi i na razvitku govora kao jednog od obilježja detekcije, a 1976. godine predstavljen je prvi prototip za prepoznavanje ljudskog govora kojeg su prvo testirale zračne snage SAD-a. Samo četiri godine kasnije, osnovana je i cijela organizacija i tijelo koje se bavi upravo proučavanjem i detektiranjem ljudskog govora u procesu identifikacije pojedinca nazvana NIST (eng. *National Institute of Standards and Technology*). Kako su se

razvijale različite metode i načini na koje se može identificirati nekoga, 1985. godine na velika vrata u upotrebu ušao je i otisak ljudskog dlana kao jedna od biometrijskih metoda. Godinu dana kasnije izdan je standard razmjene otiska prstiju koji sada čini da su sve baze međusobno djeljive i da sve policijske i obavještajne organizacije mogu pristupiti međusobnim bazama otisaka. Kroz 90-ete godine prošlog stoljeća jako se radilo na unaprjeđenju i razvitku biometrije te su se tako osnivala razna društva, organizirala natjecanja, pokretale organizacije za istraživanje svih biometrijskih metoda, a uz to razvijali su se i razni standardi, a 1977. godine prvi put je jedan takav standard postao komercijalan i dostupan svima koji su htjeli učiti i raditi na razvitku identifikacije pojedinaca. S vremenom su se počela otvarati nova učilišta i obrazovni programi na ovu temu, a 2001. godine detekcija lica prvi put je javno korištena na Super Bowl-u u svrhu identificiranja slavni i važnih ličnosti koje su posjetile stadion. S razvojem tehnologije i uređaja i biometrija je dobivala sve veći zamah, a Apple je prvi put davne 2013. godine uveo mogućnost identifikacije otiskom prsta, a nekoliko godina kasnije i prepoznavanjem lica prilikom korištenja uređaja. Same biometrijske baze, bile to baze otiska prstiju, šarenica, dlanova ili lica, postale su pozamašne, algoritmi su dovedeni do skoro nepogrešivosti, a sama biometrija uvukla se u sve aspekte svakodnevnog života ljudi [5].

2.2. Primjene prepoznavanja lica u različitim domenama

Kao što je prikazano u dijelu s povijesnim razvojem biometrike, detekcija lica je nešto novija metoda identificiranja, ali je polako ušla u veliku većinu naših skoro pa svakodnevnih aktivnosti te se koristi u mnogim područjima.

Iako smo svi navikli na otključavanje naših pametnih uređaja samo uz pomoć prikaza svog lica prednjoj kameri ili pak ulazak u urede na poslu uz pomoć skeniranja naših lica, detekcija se zapravo koristi u mnogo više područja nego smo zaista svjesni. Razne kriminalističke serije naučile su nas da je lopove ili bilo koje zlikovce moguće detektirati na raznim nadzornim kamerama u trgovinama, bankomatima i svim službenim ustanovama te na javnim površinama. U ovom dijelu teksta neće biti navedeni uobičajeni primjeri korištenja kao što su prelazak granica, sigurnosne kamere na ulicama i slično, već neki specifični i nadasve zanimljivi primjeri.

Putovanje kao jedna od najdražih aktivnosti mnogima može proći bezbolno i brzo, a može nažalost biti i neugodno iskustvo. Svakome se bar jednom dogodilo da na sigurnosnoj provjeri zapne u gužvi i jedva stigne na let. Francuska je prva pokušala riješiti taj problem uvođenjem otiska prsta. Kako bi smanjili opterećenje na pariškim aerodromima, uvedena je PARAFE (eng. *passport verification and border control system*) tehnologija otiska prstiju koja se koristi samo od strane francuske granične policije. S obzirom na to, taj je način identifikacije bio dostupan tek

4% putnika na francuskim aerodromima što nije ništa značajno smanjilo gužve i prolazak kroz sigurnosne kontrole. Kako bi i dalje unaprijedili spomenuti sustav kontrole PARAFAC, Francuzi su 2017.godine odlučili pokušati s prepoznavanjem lica kao jednom od metoda brže identifikacije putnika. To su riješili koristeći tadašnji Gemalto softver, a današnji Thales DIS softver koji je podržavao preko 105 različitih pametnih uređaja, pametnih kartica, e-tokena kao neke od raznih načina u koje je bila ukomponirana detekcija lica. Na taj su način Francuzi dobili pristup većem broju raznih baza podataka te je ujedno tako i veći broj putnika mogao koristiti ovu pogodnost pariških zračnih luka. Otisak prstiju i takve putne isprave imalo je samo 4% putnika, a putovnice s podržanim slikama lica i biometrijskim karakteristikama ima preko 50% putnika, što je prema njihovim riječima smanjilo vrijeme ukrcaja za čak 40%. Po uzoru na parišku zračnu luku, čak preko 15 različitih američkih aerodroma polako uvodi ovu metodu identifikacije na svojim prostorima, a u Europi, trenutno je ljubljanska zračna luka jedina koja uz parišku koristi ovu metodu. Proces identificiranja novih putnika zapravo je veoma jednostavan, korisnik samo mora putem pametnog uređaja predati svoju sliku, a ispostavilo se da čak 98% njih kasnije bude ispravno identificirano.

Još jedan veoma zanimljiv primjer koriste korisnici australske banke koji svojim računima na bankomatima pristupaju putem identifikacije lica umjesto korištenja bankovnih kartica. Na taj se način mogućnost korištenja tuđih kartica svela na minimum. U Australiji se identifikacija lica koristi još na jedan veoma zanimljiv način, a to je ni više ni manje pri ulasku u kockarnice i kasina. S obzirom na to da je ova ovisnost jedan od najopasnijih i najteže izlječivih, oni koji zapravo uspiju to pobijediti, zasigurno se ne žele vraćati u ovakve probleme. Upravo iz tog razloga, izlječeni ovisnici mogu dobrovoljno staviti svoje slike u bazu podataka australskih kockarnica te ako ikada pokleknju i ponovno uđu u jednu od poslovnica, bit će detektirati, osoblje će ih identificirati i ispratiti van iz kockarnice. Još jedan zanimljiv primjer dolazi opet iz Australije, a riječ je obrazovnom sustavu u osnovnim i srednjim školama. Naime, učenici mogu u školskim knjižnicama posuditi željenu knjigu, stati ispred kamere s knjigom u ruci, a sustav će ažurirati bazu tko je i kada posudio određenu knjigu. Biometrijska plaćanja su svakako neizbježan dio naše skorije budućnosti, a neki već uvelike i žive u njoj. Primjerice američki lanac brze hrane CaliBurger nudi svojim korisnicima aktiviranje korisničkog računa sa njihovim osobnim fotografijama na koje oni mogu uplaćivati iznos, a kada dođu do kioska s brzom hranom, potrebno je samo da odaberu što žele jesti, prikažu svoje lice kameri te će im automatski s korisničkog profila biti skinut onaj iznos ovisno o tome što su naručili. Na taj način smanjuju se gužve u prodavaonicama, a plaćanje postaje lako i brzo. U Kini s druge strane već skoro velika većina restorana koristi ovaj način plaćanja upravo zbog velikog zadovoljstva korisnika i pozitivnih recenzija. Kako je Kina jedna od tehnološki naprednijih, ako ne i najnaprednija

država, nije ni čudo da su razvili inovaciju u automobilske industriji gdje je sada vozačima omogućeno da svoja vozila otključavaju putem prepoznavanja lica, a ne više ključevima [6].

2.3. Proces poravnanja i detekcije lica

Iako se do sada nekoliko puta spomenuo i definirao pojam detekcije lica, za dublje i temeljitije razumijevanje istog potrebno je i jasno definirati pojam poravnanja lica i njegovu važnost u samoj detekciji. Ova dva pojma duboko međusobno su duboko povezana i zapravo ne mogu jedan bez drugoga dati zadovoljavajuće rezultate. Poravnanje lica je metodologija koja se temelji na proučavanju raznih točaka na ljudskom licu koja ga identificiraju. Danas postoje razni već istrenirani algoritmi za detekciju lica koji su dostupni na korištenje svima, no nijedan od tih algoritama nije besprijekoran te ne rješava problem loše osvijetljenih slika, krivih položaja ljudskog lica, mrlja, stranih objekata na licu i slično. Ove probleme pokušava razriješiti upravo spomenuta metoda poravnanja lica. Analizom poravnanja lica djeluje se na takozvane n točke lica koje čine glavna mjerila za detekciju lica, a to su: oči, obrve, nos, usta i čeljust [7].

Za što lakše razumijevanje povezanosti ova dva pojma, vrijeme je da se u potpunosti objasni i pojasni proces detektiranja osobe prikazan na slici 2.2 u nastavku.



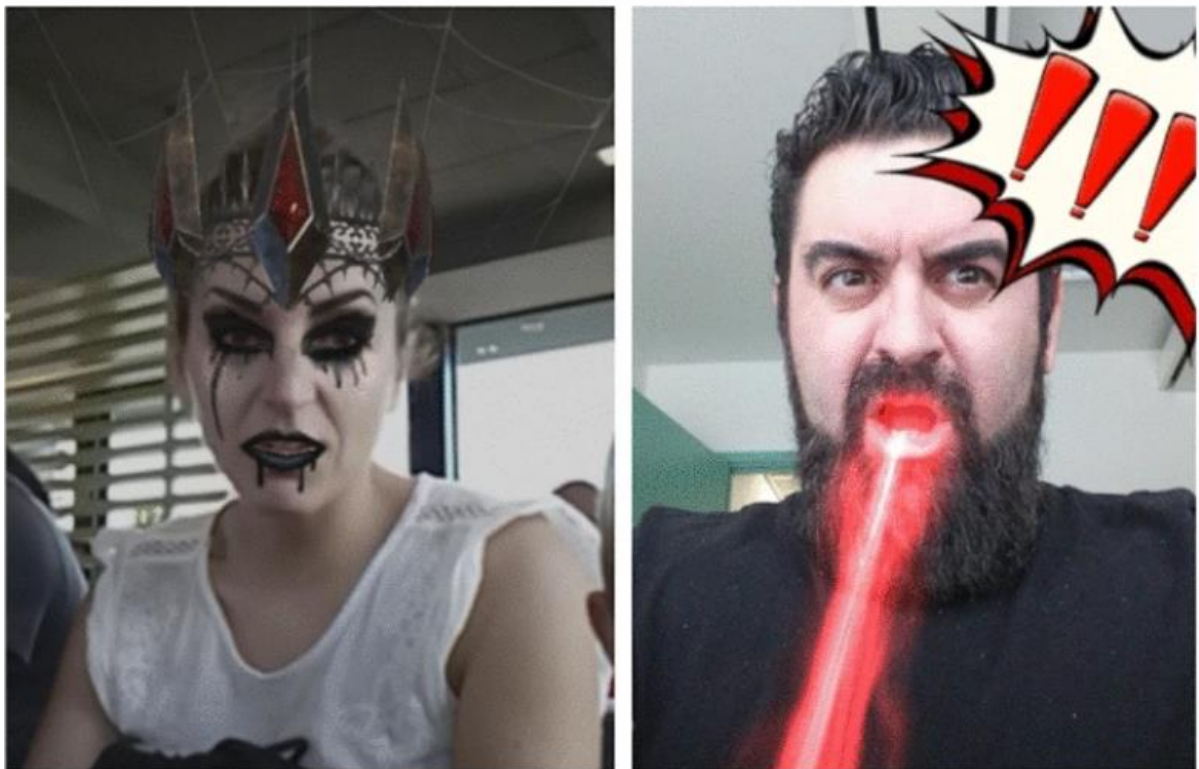
Slika 2-2 Dijagram toka detekcije lica (Vlastita izrada, 20.05.2024.)

Prvi i najlogičniji korak svakako je definiranje fotografije odnosno lica kojeg treba detektirati bilo unosom fotografije u softver, dohvatom iz baze, trenutnim skeniranjem lica ili na neki drugi način. U sljedećem koraku odvija se pretprocesiranje podataka što je zapravo naziv za pripremu slika i metapodataka da bi se s njima moglo manipulirati. Pretprocesiranje slika dijeli se u četiri različite kategorije, prva od njih orijentirana je na svjetlosti na slici i popravke oko sveukupnog osvijetljenja slike. U drugu kategoriju spadaju podaci poput geometrijskih prilagodbi slike, zatim slijedi kategorija koja za pretprocesiranje piksela uzima vrijednosti njegovih susjednih piksela.

Često se vrijednosti nekih piksela izračunavaju kao suma prethodnih te se tako analizira i izvršava pretprocesiranje slike. U posljednjoj kategoriji pretprocesa odvija se sakupljanje svih do sad dostupnih podataka o slici u svrhu obnove slike ako dođe do potrebe za prepravkom iste.

U sljedećem koraku koriste se algoritmi za detekciju lica, o kojima će više riječi biti kroz cijeli ovaj rad. Najjednostavnije rečeno, koriste se za analiziranje raznih značajki slike kako bi se uspjelo utvrditi ukoliko slika uopće sadrži područje na kojem je prikazano lice i gdje. Ukoliko sadrži, isti algoritmi služe i za lokalizaciju tih slika odnosno definiranja karakteristika kao što su gdje je točno lice sadržano te njegova površina i veličina u odnosu na cijelu sliku [8].

Sljedeći korak je korištenje algoritma za poravnavanje lica. Već je spomenuto da se metode poravnanja lica odnose na popravljavanje potencijalnih poteškoća prilikom prepoznavanja lica kao što su nagli pokreti glavom, ekstremni uvjeti osvjetljenja, raznih šumovi ako je riječ o nekim videozapisima i tako dalje. Poravnanje lica u posljednje vrijeme ima veliku ulogu u detekciji lica, ne nužno samo u video nadzornim sustavima, sigurnosnim i javnim, već se često koristi i u najobičnijim aplikacijama i svakodnevnim aktivnostima poput popularnih "selfija" na društvenim mrežama koristeći razne filtere za ljepotu ili zabavu prikazano i na slici 2.3.



Slika 2-3 Primjena algoritama poravnanja lica (Preuzeto 23.06.2024. s <https://link.springer.com/article/10.1007/s11554-021-01107-w/figures/2>)

S obzirom da se sve aplikacije i sustavi za prepoznavanje lica sastoje od raznih komponenti koje su već spomenute, poravnanje lica je jedna od važnijih, ako ne i najvažniji dio ovog procesa

pošto većina dijelova procesa ovisi upravo o poravnanju lica. Algoritmi za poravnanje dijele se u dvije kategorije: generativne i diskriminativne.

Generativne metode za poravnanje lica pokušavaju stvoriti model lica koji može mijenjati oblik, slično kao što se može raditi glinom ili bilo kojim drugim čvrstim materijalom podložnom promjenama. Ove metode koriste cjelokupne informacije o izgledu lica i obliku kako bi prilagodile model stvarnom licu. Glavni cilj je ukloniti deformacije (poput osmijeha ili podizanja obrva) kako bi se točno označile ključne točke na licu. Najpoznatije metode u ovom području su aktivni modeli oblika (ASM engl. *Algorithm State Machine*) i aktivni modeli izgleda (AAM engl. *Active Appearance Model*). ASM metoda je prva koja je uspjela točno prilagoditi model stvarnom licu. Kasnije su razvijene poboljšane metode poput algoritma SIC (engl. *Successive Interface Cancellation*), koji je vrlo precizan, ali spor, i algoritma POIC (engl. *Point of Interest Classification*), koji je brži, ali manje pouzdan. Druga grupa metoda, poznata kao metode bazirane na dijelovima, koristi lokalne dijelove lica umjesto cijelog lica. Najbolji primjer je metoda CLM (engl. *Cascade Learning Mechanism*), koja kombinira globalne obrasce lica s lokalnim informacijama oko svake ključne točke. Ova metoda je bolja za uvjete s lošim osvjetljenjem i djelomičnim preprekama na licu.

Diskriminativne metode za prepoznavanje lica koriste se za precizno određivanje položaja ključnih točaka na licu, kao što su oči, nos i usta, analizirajući slike lica. Ove metode koriste modele koji uče kako izravno povezati sliku lica s točnim koordinatama tih ključnih točaka. Postoje tri glavne vrste ovih metoda: izravne metode, koje koriste cijelu sliku lica ili njene dijelove za određivanje položaja točaka; kaskadne metode, koje počinju s grubom procjenom i postupno je poboljšavaju; te metode koje koriste duboke neuronske mreže, koje su vrlo precizne i koriste složene modele za prepoznavanje lica. Sve ove metode se obično koriste jer su vrlo točne, posebno u uvjetima kada su lica djelomično skrivena ili pod različitim kutovima. O neuronskim mrežama bit će više riječi u nastavku ovog rada [7].

2.4. Prijetnje i izazovi u detekciji i identifikaciji lica

Detekcija i identifikacija lica je veoma zanimljivo i široko područje koje privlači mnoge znanstvenike i stručnjake za istraživanje, no unatoč tomu, nisu još svi izazovi i prijetnje otklonjene prilikom detekcije. Izazova kao i u svakom drugom poslu, ima mnogo, neki od njih direktno utječu na rad modela za učenje i na krajnji ishod, neki neposredno otežavaju posao detekcije, a neki izazovi samo ukazuju na to da treba poraditi na obradi fotografija. Prvi i najveći izazov je svakako hoće li modeli i algoritmi raditi te hoće li lice na fotografiji biti ispravno

identificirano. To je veoma bitno u sigurnosnim nadzornim sustavima, policijskim i sličnim sustavima gdje je netočni identitet može ugroziti nečiju sigurnost. Prije četiri godine, istraživanja su pokazala da je stopa pogrešaka u identifikaciji samo 0,08% što znači da je napretkom i razvitkom tehnologije i korištenih metoda, danas ta stopa još niža. No na kraju ni algoritmi ni metode ne mogu ništa ako je izvorna slika loše kvalitete. Kada su ulazne fotografije koje treba obraditi u visokoj rezoluciji, s licem okrenutim direktno prema kameri, bez šumova, zamućenja i lošeg osvjetljenja, stopa pogreške je samo 0,1%. No dok se obrađuju spontane fotografije koje su snimile ljude u prolazu na ulici ili u nekim drugim lošijim uvjetima, stopa krivo ili uopće ne identificiranih slika raste na čak 9,3%. Ovako velika razlika između ova dva postotka ukazuje na izazove koji će uvijek biti problematični i na koje će se trebati što više fokusirati kod treniranja modela. To su svakako fotografije loše kvalitete, zamućene i loše osvjetljene slike, slike s licem u pokretu, slike s profilne strane lica, nagnute glave i slično. Značajke na takvim fotografijama je veoma teško detektirati, no većina metoda i algoritama koji su u nastavku opisani fokusiraju se na rješenje problema. Također jedan od izazova s kojim se cijeli svijet bori pa tako i algoritmi i modeli za detekciju lica jesu etnička, rasna i spolna ravnopravnost. U 2020. godini kod testiranja nekih modela (oni koji se koriste u komercijalne svrhe) došlo je do rezultata da će se bijela muška osoba točno identificirati s minimalnom stopom pogreške, dok je identifikacija žena i osoba drugih rasa bila puna lažno točnih rezultata. Algoritmi i modeli koji se koriste u sigurnosnim i drugim sustavi od visoke važnosti ipak ne rezultiraju toliko netočnim rezultatima [9].

Sljedeći izazov je svakako ljudska priroda u pogledu starenja i mijenjanja izgleda. Modeli mogu biti trenirani nad jednim skupom podataka, no nakon nekoliko godina, osobe s tih fotografija će ostarjeti, promijeniti izgled lica (parsing, tetovaža itd.) te ih modeli više neće točno identificirati. Rješenje ovog problema se konstantno razvija i radi se na poboljšanju, a jedan primjer i uvođenje novog modela AAM koji bi trebao i dalje uspješno izdvajati značajke sa slika istih osoba neovisno o tome koliko godina imaju na pojedinoj slici [10].

Prijetnje i rizici u tehnologiji prepoznavanja lica često se poistovjećuju, no nisu u potpunosti isti. Iako rizici sa sobom mogu donositi neke potencijalne prijetnje, trenutno u industriji postoji već neka zabrinutost oko pojedinih slučajeva kada neke situacije izazivaju prijetnje na druge sustave. Jednostavnije rečeno, prijetnje ne prijete samo modelima i algoritmima za detekciju, već raznim sustavim i državama koji ih koriste. Prva prijetnja odnosi se na sigurnost i privatnost pojedinaca. S obzirom na to da se tehnologija prepoznavanja lica koristi na više-manje svim javnim mjestima, ljudi se često mogu osjećati kao da ih se prati na svakom koraku te nikome to nije u potpunosti ugodno. Međunarodna organizacija UN čak je i u svojim poveljama navela to kao potencijalnu prijetnju za kršenje ljudskih prava. Još jedna prijetnja sustavu za detekciju lica

je čovjek. Ponekad je pitanje tko će nadgledati i upravljati sustav te trenirati modele od veće važnosti od samog načina rada sustava. Prijetnje u obliku zlouporabe podataka i curenja biometrijskih informacija o osobama nisu male i nevažne, već predstavljaju stvaran potencijalni problem [11].

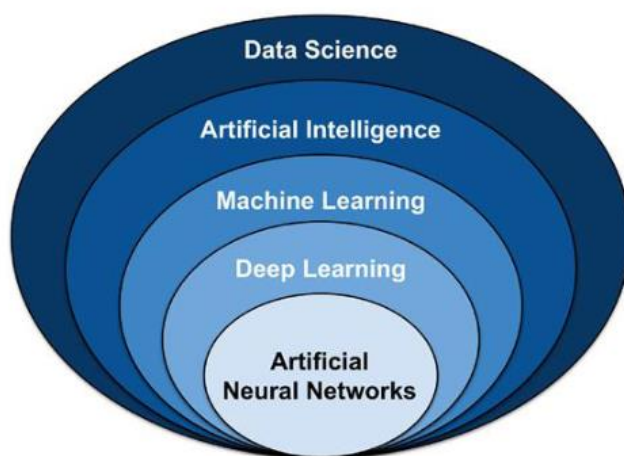
Jedna od prijetnji je svakako i krađa ili prisilno davanje identiteta u komercijalnim sustavima i aplikacijama koje koriste detekciju lica kao neki oblik autentifikacije. To se može spriječiti tako da se kao jedan dodatni korak provjere, provjeri je li osoba koja pokušava koristiti sustav živa. Na taj način smanjuje se mogućnost da je netko ukrao nečiju fotografiju i pokušava otvoriti njezin bankovni račun ili slične situacije. To se može provjeriti zahvaljujući 3D modelima koje algoritmi za detekciju podržavaju, a često se zahtijevaju zahtjevi poput toga da osoba ispred kamere trepne ili nagne glavu u određenu stranu. Također prijetnja je i zlouporaba i krađa podataka u modelima od strane onih koji modele razvijaju i treniraju [12].

Iako sustavi za detekciju i identifikaciju lica imaju mnoge izazove i moguće vanjske te unutarnje prijetnje, prednosti korištenja istih su ipak veće, a razvojem tehnologije i alata, sve od navedenog se svelo na minimalnu razinu.

3. Duboko učenje

Pojmovi poput umjetne inteligencije, strojnog učenja i dubokog učenja često se mogu čuti u medijima, no mali postotak ljudi zapravo razumije o čemu je riječ i što pojedini od tih pojmova uistinu i znači. Dok za umjetnu inteligenciju svi volimo misliti da znamo što je i da u potpunosti razumijemo njezin širok raspon, zapravo je svatko od nas daleko od razumijevanja i shvaćanja. Umjetna je inteligencija toliko složena i kompleksna da ne postoji jedna univerzalna definicija kojom bi se definirala i objasnila običnim korisnicima. McCarthy navodi da je umjetna inteligencija znanost i inženjerstvo izrade inteligentnih strojeva, posebno inteligentnih računalnih programa. McCarthy dalje definira kako je umjetna inteligencija zapravo povezana s ljudskom inteligencijom putem računala, ali sve u granicama bioloških metoda koje su vidljive i razumljive čovjeku [13].

Duboko učenje i strojno učenje često se promatraju kao cjelina jer su nesumnjivo veoma usko vezana područja, no nisu u potpunosti ista. Sa slike 3.1 u nastavku vidljivo je da su i strojno i duboko učenje zapravo jedan od dijelova koji su ključni za sastavljanje i funkcioniranje umjetne inteligencije, s time da je duboko učenje zapravo dio strojnog učenja.



Slika 3-1 Podjela podatkovne znanosti (Preuzeto 1.12.2023. s <https://tvst.arvojournals.org/article.aspx?articleid=2762344>)

Strojno učenje je zapravo područje unutar umjetne inteligencije odgovorno za razvoj algoritama koji imaju sposobnost prepoznati skup podataka i na temelju tog skupa donijeti predviđenu odluku odnosno napraviti određenu unaprijed predviđenu radnju za te podatke. Kod algoritama strojnog učenja zanimljivo je da se oni ne programiraju direktno kao ostali algoritmi

koje razvijaju programeri, već se temelje na prethodno spomenutim skupovima podataka iz kojih razvijaju metode strojnog učenja [14].

Teško je definirati duboko učenje sa samo jednom definicijom, no može se reći da je to područje temeljeno na samostalnom učenju i usavršavanju uz pomoć računalnih algoritama. Glavna okosnica dubokog učenja jesu neuronske mreže [15].

Umjetne neuronske mreže zapravo su preslika ljudskog mozga i svih neurona koji se u njemu nalaze, a glavni im je cilj simulirati klasični postupak učenja i procesiranja podataka [16].

Duboko učenje, privuklo je značajnu pažnju zbog spomenute sposobnosti analize složenih uzoraka iz velikih skupova podataka. To je zapravo jedan oblik umjetne inteligencije koji koristi navedene umjetne neuronske mreže za obradu i razumijevanje velikih količina podataka. Temeljeni na strukturi i funkciji ljudskog mozga, algoritmi dubokog učenja sastoje se od više slojeva povezanih čvorova koji obrađuju i transformiraju ulazne podatke. Ono što je potrebno naglasiti jest da duboko učenje zahvaljujući svojoj sposobnosti automatskog učenja i izdvajanja značajki, eliminira potrebu za klasičnim programiranjem kakvog poznajemo. Ova karakteristika čini ga vrlo prikladnim za zadatke poput prepoznavanja slika i govora, obrade prirodnog jezika te sustava preporuka. Također olakšava preciznije predviđanje podataka i poboljšava donošenje odluka što čini duboko učenje pogodnim za različite industrije i čini se, postat će u budućnosti, još veći dio naše svakodnevice [17].

Još jedno značajno područje utjecaja dubokog učenja leži u obradi prirodnog jezika (eng. Natural Language Processing - NLP). NLP uključuje interakciju između računala i ljudskog jezika, omogućavajući strojevima da razumiju, interpretiraju i generiraju ljudski jezik. Modeli dubokog učenja poput neuronskih mreža unaprijedili su mogućnosti NLP-a, uključujući prepoznavanje govora, strojno prevođenje, analizu i generiranje teksta. Ti modeli mogu prepoznati kompleksne jezične oblike, ali i sam kontekst nekog teksta, što dovodi do preciznije obrade jezika. Osim u obradi teksta i govora, duboko učenje pronašlo je svoj put i u obradi slika odnosno prepoznavanju i interpretiranju pojedine slike [18].

Iako je duboko učenje doživjelo brze napretke, još uvijek se suočava s izazovima koji ograničavaju njegov puni potencijal. Jedno značajno ograničenje je potreba za velikim količinama podataka za obradu modela dubokog učenja. Manjak podataka predstavlja značajan izazov, posebno u područjima gdje je stručno znanje ključno poput medicine, inženjerstva i slično. Osim toga, modeli dubokog učenja često se smatraju "crnim kutijama" zbog svoje složenosti, što otežava njihovu interpretaciju i objašnjenje. Nedostatak transparentnosti može biti problematičan, posebno u kritičnim aplikacijama poput zdravstva ili autonomnih vozila. Drugi nedostatak odnosi se na veliki broj raznih računalnih specifikacija za obradu metoda dubokog učenja. Duboke neuronske mreže na primjer zahtijevaju visoke performanse u pogledu snage i

memorije računala. To može predstavljati prepreku za pojedince ili organizacije s ograničenim resursima. Unatoč tim izazovima, postoje potencijalni napreci koji bi mogli ublažiti ove ograničenja. Transferno učenje ima veliku ulogu u tom napretku, a jednostavnije rečeno, transferno učenje je korištenje prethodno provjerenih modela učenja i njihova prilagodba novom zadatku. Ti napreci imaju potencijal značajno poboljšati praktičnost i pouzdanost tehnologije dubokog učenja. Gledajući unaprijed, očekuje se da će budući utjecaj dubokog učenja biti velik i da će još više utjecati na industrije poput zdravstva, financija i kibernetičke sigurnosti [19].

Duboko učenje spominje se jer igra važnu ulogu prilikom detekcije i identifikacije lica zahvaljujući raznim metodama i algoritmima koji se temelje na dubokom učenju, a bit će objašnjeni u ovom radu.

3.1. CNN

Konvolucijske neuronske mreže (eng. *Convolutional neural network – CNN*) važan su dio u procesu identifikacije i detekcije objekata. U ovom dijelu, CNN će se prikazati kao metoda dubokog učenja za klasifikaciju slika, ali kasnije će se koristiti i kod detekcije objekata te kod prepoznavanja lica, no neuronske mreže su mnogo više od toga.

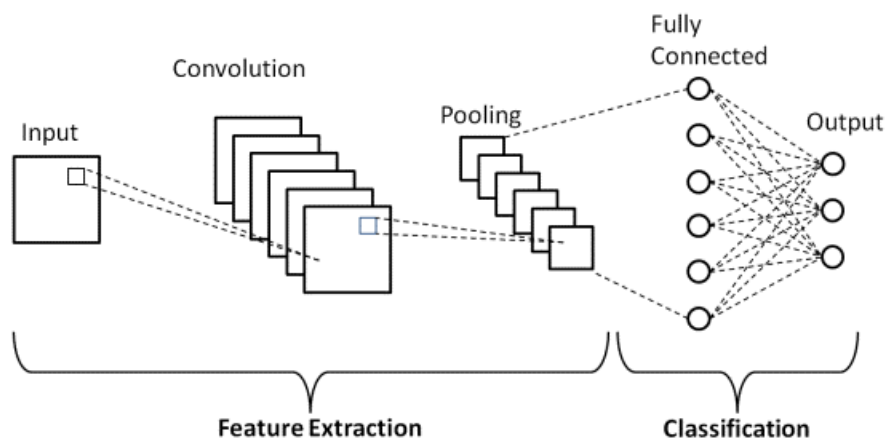
3.1.1. Arhitektura CNN-a

Konvolucijske neuronske mreže (CNN) dizajnirane su za obradu podataka nalik na mrežu kao što su slike. Ključne komponente CNN-a uključuju konvolucijske slojeve, slojeve udruživanja, aktivacijske funkcije i potpuno povezane slojeve. Konvolucijski slojevi temeljni su građevni blokovi CNN-a. Ovi slojevi primjenjuju skup filtara (također poznatih kao kerneli) na ulazne podatke kako bi izdvojili značajke kao što su rubovi, teksture, oblici i uzorci. Svaki filter klizi preko ulaznih podataka, izvodi množenje po elementima, a zatim sažima rezultate za izradu mapa značajki [20].

Sljedeći konvolucijski slojevi u okviru CNN-a su slojevi za udruživanje, od kojih se ističu maksimalno i minimalno. Maksimalno i prosječno udruživanje su tradicionalne tehnike duboko ukorijenjene u spomenute slojeve udruživanja. Operacije udruživanja pomažu pri pojednostavljenju složenih izračuna, sprječavaju potrebu da se mreže previše prilagođavaju uzorku i povećavaju mogućnost identifikacije. Potpuno povezani slojevi, poznati kao gusti slojevi, obično se nalaze na kraju CNN arhitekture. Ovi slojevi uspostavljaju veze između svakog neurona unutar jednog sloja sa svakim neuronom u sljedećem sloju, slično tradicionalnim neuronskim mrežama. Odgovorni su za kombiniranje značajki iz konvolucijskih i skupnih

slojeva, ovi slojevi oblikuju predviđanja ili klasifikacije. Oni predstavljaju završnu fazu prilikom obrade slike [21].

Slika 3.2 prikazuje osnovu arhitekturu konvolucijskih neuronskih mreža.



Slika 3-2 Arhitektura CNN-a (Preuzeto 25.02.2024. s <https://www.upgrad.com/blog/basic-cnn-architecture/>)

3.1.2. Proces treniranja

Proces obuke, ili proces treniranja označava skup nekoliko koraka koji moraju biti izvršeni kako bi se dobio krajnji rezultat. Sastoji se od prikupljanja i obrade podataka, odabira modela za detekciju lica, trening modela odnosno njegovo prilagođavanje potrebnom zadatku, validacija i testiranje performansi te analiza svega što je provedeno.

Prikupljanje i obrada podataka obično uključuje zadatke kao što je promjena veličine slika na istu veličinu, prilagodba vrijednosti piksela na određeni raspon i moguće povećanje ili smanjenje spomenutih podataka uz pomoć raznih tehnika poput rotacije, okretanja ili dodavanja šuma [22].

Nakon toga slijedi korak koji uključuje trening modela. Tijekom faze treniranja modela, CNN radi na identificiranju značajnih značajki iz ulaznih podataka te se u ovom koraku definira kako će se prikazana slika optimizirati i kako će se rizik od pogrešne detekcije svesti na najmanju razinu. Kao algoritam za trening može se koristiti propagacija unatrag (engl. *backpropagation*) koja je karakteristična za neuronske mreže. To je proces gdje se izračunavaju vrijednosti s obzirom na funkciju gubitaka za svaki parametar u mreži. Ovaj iterativni proces omogućuje mreži da prilagodi svoju težinu i u konačnici smanjuje razliku između predviđenih i stvarnih vrijednosti tijekom cijelog procesa treniranja [23].

3.1.3. Detekcija objekata

Konvolucijske neuronske mreže (CNN) igraju ključnu ulogu u detekciji lica unutar slika ili videozapisa. U detekciji lica, CNN-ovi se koriste za identifikaciju prisutnosti i lokacije lica skeniranjem ulazne slike ili video frame-a. CNN-ovi rade tako da provjeravaju sadrži li svaki okvir (eng. frame) lice ili ne, omogućujući mreži postavljanje okvira oko pronađenog lica koji će predstavljati granice za prepoznavanje i detekciju. CNN-ovi mogu učinkovito detektirati lica unutar složenih pozadina i različitih uvjeta svjetla, tame ili kvalitete slike [22].

Najpoznatiji način detekcije lica pomoću CNN-a je uz pomoć regionalnih konvolucijskih neuronskih mreža (R-CNN). R-CNN-ovi rade tako da predlože interesna područja unutar slike ili okvira, a zatim koriste CNN-ove za izdvajanje najbitnijih dijelova odnosno značajki iz tih područja. Ove se značajke naknadno unose u klasifikatore kako bi se utvrdila prisutnost lica. Također zanimljivo je da YOLO metoda i njezini algoritmi koriste u kombinaciji s CNN-ovima i tako čine još bolje i jače sustave odnosno metode za detekciju objekata kao što je lice [24].

3.1.4. Izazovi i ograničenja

Dok su konvolucijske neuronske mreže (CNN) pokazale veliki uspjeh u detekciji i prepoznavanju lica, suočavaju se s nekoliko izazova. Brze i česte promjene u uvjetima osvjetljenja predstavljaju značajnu prepreku za CNN-ove sustave detekcije i prepoznavanja lica. Promjene u razinama osvjetljenja mogu promijeniti izgled crta lica, utječući na sposobnost modela da točno identificira pojedince. CNN-ovi mogu imati problema s generalizacijom u različitim uvjetima osvjetljenja, što dovodi do manje mogućnosti prepoznavanje i detekcije. Osim toga, vanjske smetnje predstavljaju veliki izazov za CNN u zadacima prepoznavanja lica. Kada slike lica sadrže prepreke kao što su sunčane naočale, šalovi ili kosa, CNN-ovi mogu naići na poteškoće u izvlačenju značajnih obilježja i točnom podudaranju lica s licima u bazi. Iako CNN-ovi mogu naići na razne prepreke i problema, u većini slučajeva, oni su riješeni brzo i učinkovito te su zahvaljujući tome danas najtraženija metoda za detekciju objekata [25].

4. Metode dubokog učenja za detekcija objekata na slici

4.1. YOLO metoda

YOLO je poznati sustav za otkrivanje objekata u stvarnom vremenu, punog naziva You Only Look Once. Za razliku od tradicionalnih metoda otkrivanja objekata, YOLO brzo obrađuje slike u jednom prolazu kroz neuronsku mrežu, te tako poboljšava svoju sposobnost otkrivanja objekata na slikama ili videozapisima.

Na tržištu ova metoda postoji već dugi niz godina, no neki rast i početak ove metode prati se od 2015.godine kada izlazi YOLO. Na slici 4.1 ispod teksta prikazana je svaka nova verzija metode kroz godine, te je i ukratko objašnjena.



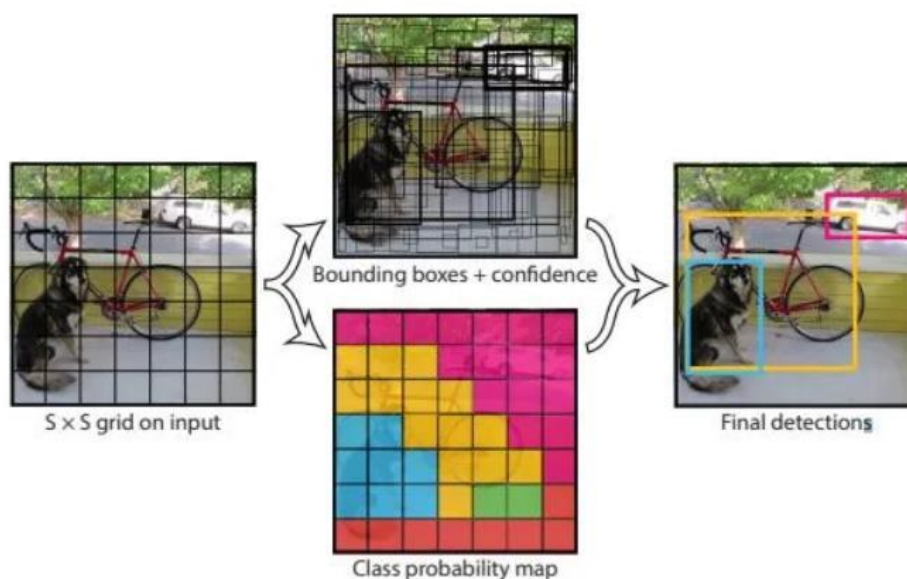
Slika 4-1 YOLO metoda kroz godine (Preuzeto 18.02.2024. s <https://www.v7labs.com/blog/yolo-object-detection#how-does-yolo-work-yolo-architecture>)

Godinu dana nakon prvog predstavljanja, YOLO algoritam dolazi u novom obliku, mnogo naprednijem od prethodnog te koji sada rješava nedostatke poput brzine i veće točnosti za razliku od prethodnika. Također u ovoj se verziji koristi i nešto što se naziva normiranje grupa, a to znači veću stabilnost i točnost pri prepoznavanju objekata. YOLO-v3 također je zanimljiva verzija koja koristi arhitekturu CNN mreža zvanu Darknet-53 koja je razvijena specifično i samo za detekciju objekata. YOLO-v4 slična je prethodnoj verziji, a razlika leži u tome da ova verzija koristi novo razvijenu CNN arhitekturu za detekciju nazvanu CSPNet (engl. *Cross Stage Partial Network*). Sljedeća verzija, YOLO-v5, koristi do tada najsloženiji arhitekturu EfficientDet, a rezultati koje daje ova metoda sada su na još višoj razini. Što se tiče brzine i točnosti, YOLO je daleko bolja metoda od konkurencije. To postiže uspostavljanjem ravnoteže između brzine i

točnosti, tako da zadovolji sve scenarije u kojima je brza analiza vizualnih podataka veoma bitna. S obzirom na spomenuto to ga čini prikladnim za situacije koje zahtijevaju trenutno donošenje odluka na temelju rezultata detekcije objekata poput pregleda kamera koje snimaju uživo. Veliku ulogu u tome ima i činjenica da YOLO procesira sliku ujednačeno, a ne pridaje važnost jednom po jednom dijelu slike. Ovaj je pristup korišten u različitim područjima kao što su autonomna vozila, sustavi nadzora i zadaci prepoznavanja slika [26].

Arhitektura YOLO-a izgrađena je uz pomoć konvolucijskih neuronskih mreža (CNN). Kada ulazna slika ulazi u mrežu, pojedinačna neuronska mreža se koristi preko cijele te slike, čime se mreži omogućuje predviđanje i prepoznavanje izravno iz cijele slike, a to je upravo bit YOLO metode – procesuiranje cijele slike odjednom [27].

U YOLO modelu, sama detekcija lica i objekata događa se zahvaljujući procesu predviđanja okvira za promatrane objekte koji bi mogli predstavljati granicu tih objekata na slici. Jednostavnije rečeno, ulazne slike se dijele u rešetku koja se sastoji od ćelija, od kojih je svaka rešetka zadužena za predviđanje granica objekata i detalja unutar slike, kao što su lica osoba. Unutar svake ćelije, YOLO predviđa određeni broj tih granica odnosno okvira, pri čemu se svakom od njih dodjeljuje određena razina koliko je ili nije pouzdana. Tijekom predviđanja graničnih okvira u YOLO-u, posebno za crte lica ili druge vrste objekata, algoritam radi na procjeni koordinata za svaki taj okvir koji se odnosi na pojedinu ćeliju mreže. Predviđene koordinate obično uključuju središnje koordinate tog okvira, njegovu širinu, visinu i rezultat pouzdanosti koji prikazuje vjerojatnost da okvir sadrži predmet odnosno objekt ili u ovom slučaju lice osobe koje tražimo. Na slici 4.2 u nastavku nalazi se grafičko objašnjenje spomenutih granica, rešetki i okvira.



Slika 4-2 YOLO mreža rešetki i graničnih okvira (Preuzeto 19.02.2024. s <https://towardsdatascience.com/evolution-of-yolo-yolo-version-1-afb8af302bd2>)

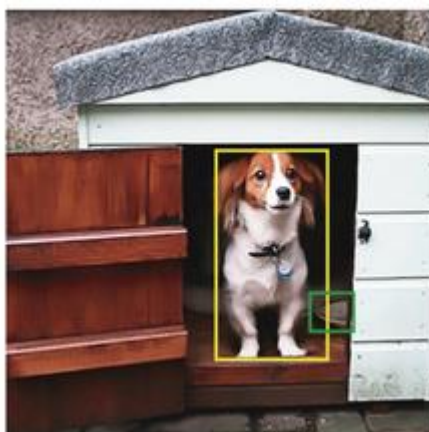
Kako bi se ova metoda još bolje prikazala, biti će ukratko prikazana svaka verzija YOLO metode.

4.1.1. YOLOv1

U jednoj od prvih verzija YOLOv1 svi se potrebni koraci odvijaju odjednom. Prvo, YOLO podijeli ulaznu sliku u mrežu od $S \times S$ kvadrata. Naprimjer ako je S proizvoljni broj 5 onda podijeli u mrežu od 5×5 odnosno 25 kvadrata. Svaki kvadrat predviđa B (proizvoljan broj) okvira (pravokutnika) koji mogu sadržavati objekte, pri čemu svaki okvir ima pet vrijednosti: pouzdanost (koliko je model siguran da okvir sadrži objekt i koliko je okvir precizan), koordinate središta okvira u odnosu na kvadrat, te visinu i širinu okvira u odnosu na cijelu sliku. Uz to, za svaki kvadrat model predviđa pouzdanost za C različitih klasa objekata (npr. automobili, ljudi, psi). Izlaz iz YOLO modela je višedimenzionalna matrica dimenzija $S \times S \times (B \times 5 + C)$. Na primjer, za $S = 7$, $B = 2$ i $C = 20$, izlaz bi bio dimenzija $7 \times 7 \times 30$. Nakon što model generira okvire, koristi se tehnika potiskivanja preklapajućih graničnih okvira (engl. *non-maximum suppression*, NMS) kako bi se uklonili višestruki okviri i zadržali samo najbolji okvir za svaki

objekt. Najjednostavnije rečeno, YOLOv1 dijeli sliku u nekoliko velikih kvadrata prema kojima onda provodi detekciju objekata odnosno lica.

U ovoj se verziji koristi 24 konvolucijska sloja. Konvolucijski sloj je zapravo građevni dio konvolucijske neuronske mreže. Konvolucija u pogledu matematičke operacije predstavlja onu operaciju koja se izvršava nad neke dvije funkcije, a kao rezultat daje potpuno novu treću funkciju [28]. Arhitektura YOLOv1 sastoji se od tih spomenutih 24 konvolucijska sloja od kojih posljednja dva zapravo predviđaju okvire unutar kojih bi se mogao detektirati neki objekt, a o kojim okvirima je riječ može se vidjeti na slici 4.3 u nastavku.



Slika 4-3 YOLOv1 primjer arhitekture (Preuzeto 28.06.2024. . Terven ,Cordova-Esparanza 2023.)

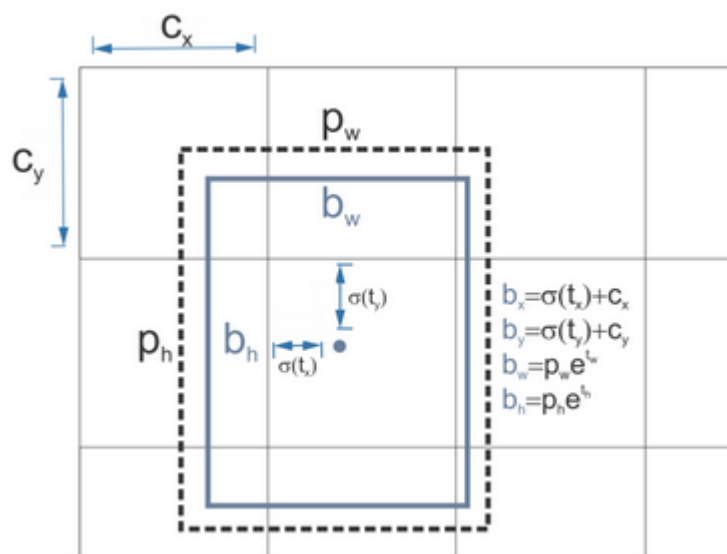
Kao i svaki drugu metodu za detekciju objekata, i YOLOv1 treba trenirati i ispravljati kako bi dao najbolje moguće rezultate. Treniranje YOLO metode temelji se na funkciji gubitaka čiji je glavni cilj usporedba stvarnih i predviđenih parametara, a važno je napomenuti da će detektirati grešku samo ako unutar okvira postoji objekt. Unutar funkcije gubitka može se definirati nekih pet parametara, prvi parametri predstavljaju grešku u predviđenoj lokaciji i veličini samih okvira, treći parametar odnosi se na grešku pouzdanosti kada je objekt detektiran unutar okvira, a četvrti parametar kada objekt nije detektiran. Peti parametar je gubitak klasifikacije koji mjeri koliko je model pogriješio u predviđanju klase objekta, a klasa objekta je najjednostavnije rečeno neka kategorija kojoj pripisujemo taj objekt (na slici iznad klasa bi bila pas). Ovo se računa samo ako se u ćeliji zaista nalazi objekt [29].

4.1.2. YOLOv2

S novim verzijama najčešće uvijek dolaze nova poboljšanja, nove naprednije i bolje značajke te razne druge izmjene. Isto tako dogodilo se i s novijom verzijom YOLO metode, YOLOv2.

Ova je verzija razvijena od strane dvojice znanstvenika, Redmonda i Farhadija 2017. godine kada je i obavljena na jednoj od najpoznatijih računalnih konferencija, CVPR.

YOLOv2 također ima određena poboljšanja u odnosu na prvu verziju. Tako je s YOLOv2 omogućeno korištenje slika u većoj rezoluciji. Dok je s prvom verzijom bilo moguće koristiti modele s rezolucijom 224 x 224, YOLOv2 omogućio je da se to digne na višu razinu i da se koristi rezolucija 448 x 448. U ovoj verziji uveo se i novi pojam sidrenih okvira (engl. *anchor box*) koji označava unaprijed definirane oblike okvira koji bi mogli odgovarati obliku pojedinog objekta kojeg je potrebno detektirati, primjerice postojali su posebni sidreni okviri za detekciju ljudi, ljudskih lica, životinja ili bilo kojeg drugog objekta kojeg se pokušavalo detektirati prilikom treniranja metode. Svaka ćelija mreže ima više sidrenih okvira, a sustav predviđa koordinate i klasu za svaki okvir, što znači da veličina izlaza mreže ovisi o broju sidrenih okvira po ćeliji. Za predviđanje lokacija okvira, YOLOv2 nije koristio metode koje predviđaju pomake, već je slijedio isti princip kao i YOLOv1, predviđajući koordinate okvira u odnosu na ćeliju mreže. Mreža predviđa pet okvira po ćeliji, svaki s pet vrijednosti. Takva se nova arhitektura naziva Darknet-19 jer koristi 19 slojeva konvolucijske mreže prikazano i na slici 4.4 ispod teksta.



Slika 4-4 YOLOv2 arhitektura (Preuzeto 21.06.2024. Terven ,Cordova-Esparanza 2023.)

4.1.3. YOLOv3

Nastavno na YOLOv2, nova metoda dubokog učenja priključila se detekciji lica 2018. godine kao treća verzija YOLO metode. U prethodnom dijelu spomenut je novi pristup s takozvanim sidrenim okvirima, a treća verzija unaprjeđuje ovaj pristup na način da se predviđa prisutnost

objekta u okviru u obliku ocjena 1 i 0 gdje jedan označava istinu odnosno prisutnost objekta, a 0 laž tj. prazan okvir. Sljedeća razlika u odnosu na prethodnu verziju omogućuje dodjelu više oznaka (klasa) istom okviru, što se može dogoditi kod složenijih skupova podataka s klasama koje se preklapaju. Na primjer, isti objekt sada može biti označen kao "životinja" i "pas". U YOLOv3 razvija se i drugačija arhitektura nazvana Darknet-53. Mreža se sada sastoji od 53 konvolucijska sloja, a svaki od njih koristi takozvanu normalizaciju serije (engl. *batch normalization, batch norm*) i Leaky ReLU aktivacijsku funkciju za treniranje neuronskih mreža [29].

Normalizacija serije je metoda koja stabilizira i ubrzava treniranje neuronskih mreža tako što normalizira ulaze svakog sloja. To znači da svaka grupa podataka (batch) prolazi kroz proces gdje se standardiziraju vrijednosti, što pomaže u održavanju stabilnog protoka podataka kroz mrežu, a na taj način treniranje postaje točnije i otpornije na pretreniranje (engl. *overfitting*) [30]. Leaky ReLU aktivacija je također način treniranja mreža koji za razliku od klasične ReLU funkcije na kojoj se temelji i koja svim negativnim ulazima u funkciju postavlja vrijednost na 0, Leaky ReLU daje malenu vrijednost (npr. 0,01x) svim negativnim ulazima umjesto nule. To omogućuje prolazak malih negativnih signala, što pomaže u izbjegavanju problema neaktivnih neurona, gdje neuroni prestaju učiti jer gradijenti imaju vrijednost nula [31].

Uz potpuno unaprijeđenu arhitekturu, YOLOv3 donosi i još jednu novost, a to je višesmjerna predikcija. Sada je moguće predviđanje postojanja objekta izvršiti nad više okvira odjednom, tako se riješio jedan od važnijih nedostataka svih prethodnih YOLO verzija, a to je uspješna detekcija manjih objekata unutar okvira.

4.1.4. YOLOv4

YOLOv4 pojavio se dvije godine kasnije i to ne od istih autora koji su do sad razvijali tu metodu, no nikakve značajne promjene od dosadašnje YOLO metodologije nisu učinjeni, već samo napredci i pokušaji poboljšavanja performansi. Razvijena su dva nova smjera koja trebaju donijeti poboljšanje, a to su promjene strategije treniranja modela koja donosi nešto povećani trošak, ali vrijeme treniranja ostaje otprilike isto. Drugi smjer odnosi se na poboljšanje točnosti, poput povećavanja vidnog polja i kombiniranja raznih značajki treniranja, što sa sobom donosi značajni promjene u troškovima treninga. Dodatnim istraživanjem i raznim pristupima, ova se verzija prilikom detekcije i dalje oslanja na sidrene okvire iz prethodne verzije te tu nije došlo do nikakvih značajnijih promjena. Ispitujući prethodne arhitekture, Darknet-53 arhitektura iz YOLOv3 verzije također se pokazala najboljom uz male preinake i poboljšanja pa tako u ovoj verziji nastaje nova arhitektura nazvana CSPDarknet53-PANet-SPP. Neke od novosti su također

korištenje DropBlock (umjesto Dropout) metode za regularizaciju metode koja je prvenstveno razvijena za konvolucijske neuronske mreže, a bolje sprečava pretreniranje jer više ne isključuje neurone pojedinačno već u manjim količinama odnosno više njih u mini serijama.

YOLOv4 također uvodi novu tehniku koja umanjuje preveliku sigurnost modela u svoje predikcije. Umjesto dodjeljivanja potpune sigurnosti ispravnoj klasi (vrijednost 1 odnosno istina) i vrijednost 0 tj. promašaj, ova tehnika dodjeljuje vrijednosti koje su malo manje od 1 za ispravnu klasu i obrnuto s obzirom na to da ne može sve biti tako crno i bijelo odnosno da objekt 100% (ne) postoji unutar okvira. To pomaže modelu da ne bude previše siguran u svoje predikcije. U prethodnoj verziji objašnjeno je što je normalizacija serije, a YOLOv4 sa sobom donosi unaprijeđenu i drugačiju normalizaciju -CmBN (engl. *cross mini-batch normalization*). Ova normalizacija prikuplja statistike iz cijele serije podataka, a ne iz pojedinačnih mini serija kao normalizacije u YOLOv3 verziji. To omogućuje stabilnije i točnije procjene parametara, što poboljšava performanse modela, osobito kada se trenira na malim serijama podataka.

4.1.5. YOLOv5

YOLOv5 dolazi u pet modificiranih verzija: YOLOv5n (mini), YOLOv5s (malo), YOLOv5m (srednje), YOLOv5l (veliko) i YOLOv5x (extra veliko). Razlike su u širini i dubini konvolucijskih dijelova te su prilagođene određenim aplikacijama i zahtjevima hardvera. Na primjer, YOLOv5n i YOLOv5s su lagane verzije namijenjene uređajima s manje resursa, dok je YOLOv5x optimiziran za visoke performanse uz određeni gubitak brzine.

Najnovija verzija YOLOv5 (v7.0) uključuje verzije koje podržavaju klasifikaciju i segmentaciju objekata. YOLOv5 je projekt otvorenog koda koji korisnici mogu lako koristiti, trenirati i implementirati, a nudi i mobilne verzije za iOS i Android te razne integracije za označavanje, treniranje i implementaciju.

4.1.6. YOLOv6

Nešto novija verzija pojavljuje se u rujnu 2022. godine, a za razliku od svojih prethodnika, ova verzija daje superiornije rezultate u pogledu brzine i preciznosti detekcije. YOLOv6 nudi osam modela, od YOLOv6-N do YOLOv6-L6. Prilikom ispitivanja i testiranja, najbolji model postigao je točnost od 57,2% pri otprilike 29 sličica po sekundi. Ovo pokazuje sposobnost YOLOv6 modela da pruži visoke performanse i jako dobre rezultate u kratkom razdoblju.

4.1.7. YOLOv7

Iste godine kao i prethodna verzija, predložena je i verzija YOLOv7. Promjene u arhitekturi svakako su donijele pozitivne rezultate te je do tada YOLOv7 metoda imala najbolje rezultate. Počinje se koristiti nova strategija učenja ELAN koja omogućuje da modeli bolje uče tako da se usmjeravaju prema gradijentnim putevima. Gradijenti su matematički vektori koji pokazuju smjer i brzinu najbržeg porasta neke funkcije. U kontekstu dubokog učenja, gradijenti se koriste za optimizaciju modela tijekom učenja. Oni predstavljaju smjer u kojem treba prilagoditi parametre modela kako bi se smanjila greška predviđanja odnosno već objašnjena funkcija gubitka. YOLOv7 je predložio novu strategiju za prilagodbu različitih karakteristika modela kako bi se postigle određene performanse ili zahtjevi u vezi s hardverom ili primjenom koja se temeljeni na tome da se dubina i širina bloka skaliraju istim faktorom kako bi se očuvala optimalna struktura modela i smanjili napori i performanse hardvera.

4.1.8. YOLOv8

YOLOv8 dolazi kao sljedeća verzija s većim promjenama, a koja opet ne odstupa previše od cjelokupne arhitekture YOLO metode. Temelji se većinom na v5 verziji s naravno nekim promjenama i izmjenama u arhitekturi. Koristi se dizajn koji omogućuje modelu da neovisno obrađuje zadatke kao što su prepoznavanje objekata, klasifikacija i precizno označavanje mjesta objekata. To znači da svaka komponenta modela radi na svoj način kako bi donijela točniji rezultat. Poseban dio YOLOv8 koristi se za semantičku segmentaciju. To znači da model može razumjeti i označiti različite dijelove slike kao što su ceste, nebo ili druge pozadinske informacije. Ova se verzija može lako koristiti putem naredbenog retka ili instalirati kao PIP paket. Također dolazi s mnogim alatima za označavanje, treniranje i implementaciju, što olakšava rad s modelom [29].

4.1.9. YOLOv9

Iako je YOLOv8 predstavljena kao posljednja službena verzija metode, u veljači 2024. godine, predstavljena je još jedna novija, YOLOv9 verzija čiji su autori znanstvenici Chien-Yao Wang, I-Hau Yeh i Hong-Yuan Mark Liao. Trenutno YOLOv9 podržava samo detekciju objekata na slikama, a ne i lica i slično. U svojoj arhitekturi donosi novi koncept PGI (engl.

programmable gradient information). Najveća prednost ovog koncepta je što omogućuje odabir proizvoljne funkcije gubitka koja bi mogla biti najbolja za ostvarivanje cilja koji je što točnije i brže treniranje modela. Također, PGI omogućuje i primjenu u neuronskim mrežama odnosno u proizvoljnim veličinama blokova neuronskih mreža što znači da korisnik može prilagoditi svoj blok na kojem testira prema uređajima koje koristi s obzirom na softverske i hardverske specifikacije tih uređaja. Ovaj novi model donosi sa sobom mnoge prednosti u pogledu brzine i točnosti, spomenutog PGI koncepta koji pak ima svojih pozitivnih aspekata te koji u kombinaciji s novo razvijenom neuronskom mrežom GELAN (engl. *generalized efficient layer aggregation network*) ostvaruje jako dobre rezultate na raznim testiranjima i treninzima modela. YOLOv9 već se uvelike koristi u industriji, u autonomnim vozilima, prilikom sportskih analiza i brojanja ljudi na velikim događanjima, a jedan od primjera primjene nalazi se i na slici ispod gdje se uz pomoć YOLOv9 modela može unaprijediti ulična detekcija i detekcija u gradskim videonadzorima [32].

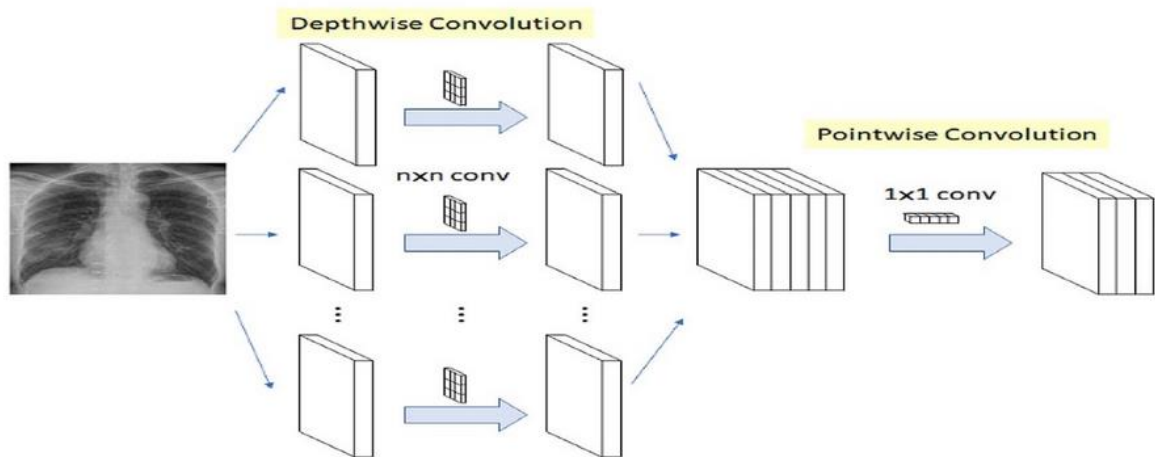


Slika 4-5 Primjer korištenja YOLOv9 modela (Preuzeto 23.06.2024. s <https://viso.ai/computer-vision/yolov9/>)

4.2. MobileNet

Uz YOLO metodu, postoje još mnogo raznih metoda i modela koji se koriste u dubokom učenju prilikom detekcije objekata, a jedan od njih je i MobileNet. Razvijen je 2017. godine od strane Google-ovih stručnjaka i znanstvenika prvenstveno namijenjen za manje uređaje kao posljedica sve veće uporabe pametnih telefona i uređaja. S obzirom na to da se većina toga vezanog uz detekciju objekata oslanja na neuronske mreže koje same po sebi zahtijevaju velike resurse u pogledu hardverskih performansi uređaja, MobileNet je uvođenjem drugačije i naprednije arhitekture prilagodi korištenje konvolucijskih neuronskih mreža te manjim i slabijim uređajima [33].

Arhitektura ovog modela oslanja se na već nekim spomenutim i objašnjenim pojmovima kao što su normalizacija serije skupa podataka te ReLU funkcije za treniranje modela. Primarno na čemu se temelji cijeli model je korištenje dubinski odvojenih konvolucija, a cijelo učenje razvio je Googleov stručnjak Francois Chollet te se naziva Xception [33]. Klasični konvolucijski slojevi prilikom treniranja modela u 3D prostorima koriste jednu jezgru za učenje o širini, visini i međuprostornim kanalima koji povezuju širinu i visinu i to svaka jezgra uči zasebno. Xception model dubinski odvojive konvolucije (engl. *depthwise separable convolution*) koje se sastoje od dubinske (engl. *depthwise convolution*) i točkaste (engl. *pointwise convolution*). Dubinska konvolucija omogućuje detaljniji rad sa slikom s obzirom na to da za svaki ulazni kanal primjenjuje jedan zasebni filter. Tako primjerice ako postoji slika s tri kanala boja (npr. crvena, zelena i plava), konvolucija po dubini omogućuje da se svakom kanalu pridoda njegov filter što omogućuje detaljniji, točniji i bržu konvoluciju tih podataka. Točkasta konvolucija koristi se kasnije te spaja sve obrađene filtere u jednu izlaznu mapu. Taj je model i prikazan na slici 4.6 gdje se svakoj ulaznoj točki slike pridao njezin filter, a kasnije su svi uz pomoć točkaste konvolucije spojeni u završni zajednički prikaz [35].



Slika 4-6 Model dubinski odvojive konvolucije (Preuzeto 25.06.2024. s <https://viso.ai/deep-learning/xception-model/>)

Xception model sastoji se od tri dijela: ulazni, srednji i izlazni. U ulaznom dijelu ulazi slika dimenzija 299×299 piksela i s 3 kanala (RGB). To znači da je slika obojena i sastoji se od crvene, zelene i plave komponente. Nad slikom se koristi konvolucijski sloj s 32 filtera veličine 3×3 i s pomakom 2×2 . Na kraju procesa dolazi ReLU funkcija koja preslikava samo pozitivne vrijednosti, dok negativne stavlja u 0 (izlaz je maksimum od 0 i ulaza).. Nakon što su sve značajke slike izdvojene provodi se takozvana dubinski odvojiva konvolucija što je zapravo tehnika koja razdvaja dubinsku konvoluciju (ona što se provodi za svaki kanal zasebno) i točkastu konvoluciju koja kombinira sve te kanale zajedno s ciljem smanjenja nepotrebnih matematičkih operacija koje opterećuju sustav. Središnji dio Xception modela ponavlja se osam puta, a svako ponavljanje sastoji se od dubinski odvojene konvolucije sa 728 filtera i ReLU funkcijom. Izlazni dio modela omogućuje da kombinacija svih gore navedenih komponenti omogući mreži da izvuče složene značajke iz slika, sažme te značajke u kompaktni vektor i zatim klasificira slike s visokom točnošću [34].

Nakon razumijevanja Xception modela, arhitektura i način rada MobileNet metode postaje jasnija s obzirom na to da se temelji na spomenutom modelu. Baš kao i YOLO metoda, i MobileNet se razvijao u nekoliko verzija od kojih se svaka po nečemu razlikuje, a objašnjene su svaka zasebno u sljedećim odlomcima.

Što se tiče primjene MobileNet metode, ona je svakako raširena u mnogim područjima, ali u zdravstvu je ostavila najznačajniji doprinos. U doba globalne pandemije koronavirusom, znanstvenici iz svih područja pokušavali su dati svoj doprinos da se čim prije dođe do točnijeg uzroka, a zatim i rješenja tadašnjeg problema s virusom. U području računalnog vida, MobileNet je svakako donio zavidne rezultate pred drugim metodama te je zahvaljujući svojoj naprednoj, a ujedno jednostavnoj arhitekturi, uspio dati svoj doprinos u godinama pandemije. Za detekciju

virusa u tijelu, koristila se verzija MobileNetV2, a rendgenska slika pluća mogla se svrstati u tri kategorije (zdrava pluća, Covid-19 pluća i druge plućne bolesti) zahvaljujući ovoj metodi dubokog učenja za detekciju. Također s obzirom na obavezno nošenje maske za lice u tom razdoblju, MobileNet pokazao se kao dobra metoda za detekciju da li osoba nosi masku i je li to na ispravan način. Na 9000 testnih slika, metoda je točno prikazala 87.96% detektiranih maski, a čak 93.5% točnosti je li maska postavljena ispravno ili nije. No to nije jedini primjer korištenja MobileNet-a u zdravstvu. Naime uz pomoć ove metode moguće je detektirati da li neka kožna oštećenja i anomalije upućuje na benigne ili pak maligne bolesti kože. Riječ je o mobilnim aplikacijama koje primaju slike kože, a zatim mogu vratiti rezultate koji su čak do 90% točni. Jedan takav primjer korištenja te aplikacije vidljiv je na slici 4.7.



Slika 4-7 Primjer korištenja MobileNet metode u zdravstvu (Preuzeto 25.06.2024. s <https://viso.ai/deep-learning/mobilenet-efficient-deep-learning-for-mobile-vision/>)

S obzirom na to da je dokazano da dobro djeluje u zdravstvu i otkrivanju bolesti na temelju slika, isto tako značajnu ulogu ima i u botanici gdje je također uz pomoć raznih aplikacija koje koriste MobileNet moguće detektirati bolest lišća biljaka, odnosno u jednom radu može detektirati 10 različitih bolesti lišća rajčice [33].

Spomenuto je da postoji nekoliko verzija MobileNeta kako se razvijao, a prikazane su ukratko u sljedećem dijelu rada. S obzirom da je u općenitom dijelu o MobileNet metodi objašnjena zapravo i prva verzija, ostatak će se odnositi na preostale verzije.

4.2.1. MobileNetV2

MobileNetV2 predstavlja napredak u odnosu na izvorni MobileNetV1 u pogledu sposobnosti da postigne dobar balans između veličine modela i točnosti, što ga čini idealnim za uređaje s ograničenim resursima. Iako i prethodna verzija dobro funkcionira na manjim uređajima, u verziji broj dva to je još više istaknuto i navedeno kao prednost [35].

Sama arhitektura MobileNetV2 ne odstupa toliko daleko od Xception modela koja se koristi pri izgradnji MobileNetV1. I dalje se koristi konvolucijski sloj s 32 filtera te ReLU. Točnije sada se koristi verzija ReLU6 (ReLU limitirana na maksimalni izlaz 6) baš namijenjena za što veću točnost kod mobilnih uređaja. Prva razlika između ova dva modela je da se u ovoj verziji koristi neprestano povećanje broja kanala koji sudjeluju u procesu. Proces kojim se povećava taj broj kanala (riječ je o istim onim kanalima opisanim u MobileNetV1 verziji) naziva se *bottleneck* blok "usko grlo" (engl. *bottleneck*), ili blok inverznih reziduala (engl. *inverted residual*). Blok "usko grlo" u neuronskim mrežama označava upravo ono što i znači u doslovnom prijevodu – grlo boce, ali u ovom slučaju usko grlo kroz koje prolaze informacije u neuronskim mrežama. Blok "usko grlo" je zapravo više tanjih slojeva mreže koji zajedno smanjuju veličinu podataka koja prolazi kroz njih. U MobileNetV2 arhitekturi blokovi "usko grlo" sastoje se od 1x1 (točkastog) konvolucijskog sloja za ekspanziju, sloja dubinske konvolucije objašnjene ranije i ponovo 1x1 konvolucijskog sloja za projekciju (na broj izlaznih kanala) [37].

Arhitektura MobileNetV2 sastoji se od nekoliko slojeva prikazanih na sljedećoj slici 4.8.

Layer Type	Input Size	Output Size	Kernel Size	Stride	Expansion Factor
Initial Conv	224x224x3	112x112x32	3x3	2	–
Inverted Residual Block	112x112x32	112x112x16	3x3	1	1
Inverted Residual Block x2	112x112x16	56x56x24	3x3	2	6
Inverted Residual Block x3	56x56x24	28x28x32	3x3	2	6
Inverted Residual Block x4	28x28x32	14x14x64	3x3	2	6
Inverted Residual Block x3	14x14x64	14x14x96	3x3	1	6
Inverted Residual Block x3	14x14x96	7x7x160	3x3	2	6
Inverted Residual Block x1	7x7x160	7x7x320	3x3	1	6
Final Conv	7x7x320	7x7x1280	1x1	1	–
Global Avg Pooling	7x7x1280	1x1x1280	–	–	–
Fully Connected	1x1x1280	1x1x1000	–	–	–

Slika 4-8 Slojevi MobileNetV2 (Preuzeto 25.06.2024. s <https://www.geeksforgeeks.org/mobilenet-v2->

Prvi sloj je onaj već spomenuti s 32 filtera i jezgrom 3 x 3. Nakon njega slijedi nekoliko *bottle neck* blokova slojeva mreže, zatim finalni konvolucijski sloj koji ih sve spaja u jedan. MobileNet V2 postigao je dobar balans između točnosti i učinkovitosti, što ga čini idealnim za mobilne aplikacije. To znači da može brzo i točno obraditi podatke, dok koristi minimalnu količinu resursa (kao što su procesor i memorija). Arhitektura se može lako prilagoditi kako bi zadovoljila specifične potrebe različitih aplikacija podešavanjem mreže za rad s različitim veličinama ulaznih podataka ili za postizanje različitih razina točnosti tih podataka. U usporedbi sa svojim prethodnikom, MobileNetV2 pruža bolje performanse s manje parametara i nižim manjim vremenom proračuna. To znači da je sposoban postići bolje rezultate, dok koristi manje resursa i vremena za obradu. Koristi se u raznim situacijama i područjima, a najveću primjenu ova je metoda doživjela u detekciji objekata, obradi fotografija, u područjima računalnog vida u IoT uređajima i slično.

4.2.2. MobileNetV3

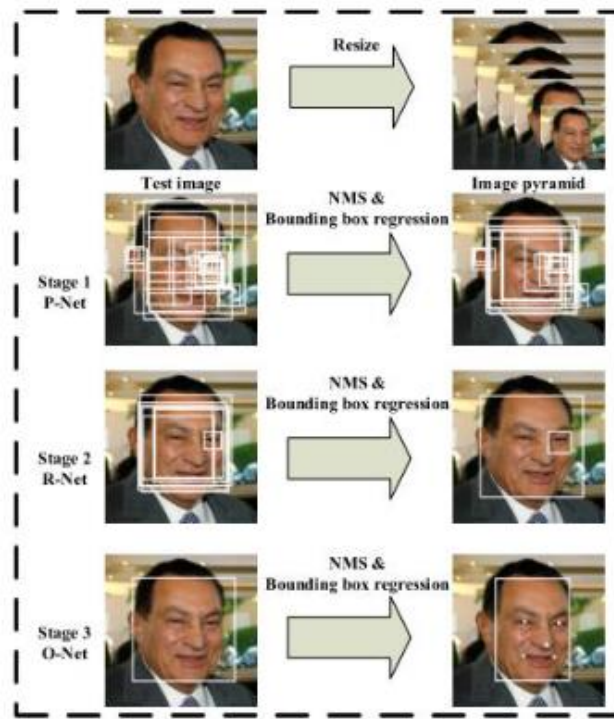
MobileNetV3 model pojavio se 2019. godine, a baš kao i sve njegove prethodnike, razvila ga je grupa stručnjaka u Googlu. Može se reći da se dijeli na neka dva osnovna modela: veći i manji MobileNetV3. Zovu se veći i manji jer su vezani na slučajeve s većim ili manjim udjelom resursa koji su potrebni za obavljanje procesa. Veći MobileNetV3 više se ne koristi primarno za manje uređaje kao što su mobiteli, već je primjenjiv na velike i složene skupove podataka, dok je manji model i dalje primarno namijenjen radu mobitela, tableta, iPada i sličnih uređaja [36].

Kako bi se to omogućilo, arhitektura ove verzije morala je biti nešto drugačija od prethodnih. MobileNetV3 je konvolucijska neuronska mreža koja je prilagođena CPU-u mobilnih telefona kombinacijom pretraživanja mrežne arhitekture uz znanje o korištenom hardveru, nadopunjena algoritmom NetAdapt, a zatim poboljšana kroz nove arhitekture. Napredak uključuje komplementarne tehnike pretraživanja, nove učinkovite verzije nelinearnosti praktične za mobilne postavke i novi učinkoviti dizajn mreže [37].

4.3. MTCNN

Nakon što su u prethodnom dijelu konvolucijske neuronske mreže konačno i detaljno objašnjene, vrijeme je da se prikaže jedna pod-varijanta tih mreža nazvana MTCNN, a to je zapravo skraćena za engleski naziv *Multitask cascaded convolutional neural network*.

MTCNN metoda razvijena je 2016. godine, a danas je jedna od često korištenih metoda za detekciju lica na slikama. Sastoji se od tri dijela: detekcije područja koje bi moglo sadržavati lice ili željeni objekt, usavršavanje položaja i na kraju položaj lica i 5 značajki lica. Na slici 4.11 prikazane su tri faze MTCNN-a. U prvoj fazi koja se naziva P-Net uz pomoć istoimene mreže, detektiraju se potencijalni okvir slike unutar kojih bi moglo biti podržano lice. Zatim slijedi faza R-Net u koju ulaze samo oni okviri koji su u prethodnoj fazi prošli test da sadrže lice. Uz pomoć R-Net mreže procjenjuje se i poboljšava točna lokacija okvira dobivenih u prvoj fazi što naposljetku omogućuje pronalazak jasnije i bolje lokacije lica. U posljednjoj, trećoj fazi koja se naziva O-Net faza, stvaraju se konačni konkretni okviri oko lica koji ukazuju na skoro pa 100% točnost da unutar tog okvira uistinu i postoji lice te da je ispravno detektirano [38].



Slika 4-9 Prikaz radnog slijeda MTCNN-a (Preuzeto 27.06.2024. s

https://jasonyanglu.github.io/files/lecture_notes/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0_2020/Project/Face%20Recognition%20Based%20on%20MTCNN%20and%20FaceNet.pdf)

5. Metode dubokog učenja za klasifikaciju slika

U nastavku će biti prikazane dodatne metode koje se koriste prilikom klasifikacije slika.

5.1.1. ResNet

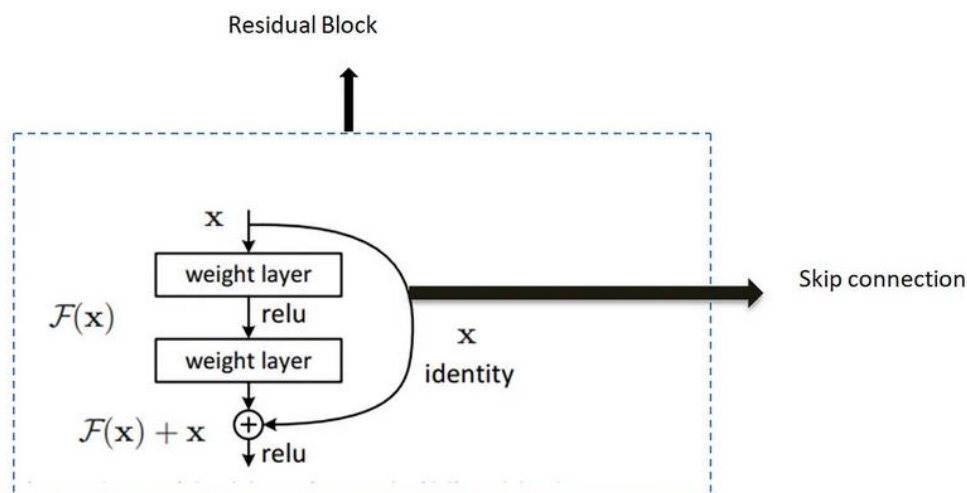
ResNet, punog imena *Residual Neural Network*, je model dubokog učenja izvorno predložen za klasifikaciju slika, razvijen 2015. godine od autora Kaiming He, Xiangyu Zhang, Shaoqing Ren te Jian Sun. ResNet je za svoj tako dobar uspjeh i širok domet morao riješiti neke poteškoće koje su do tad postojale u dubokim neuronskim mrežama. Prva prepreka na koju su autori ove metode naišli je preveliko smanjenje gradijenta koji se koriste u funkcijama gubitka. Do toga je dolazilo zbog velikog broja slojeva neuronskih mreža koji se kako ide treniranje i povećavaju, a gradijenti ne mogu pratiti tolike slojeve te nisu mogli model trenirati ispravno i podjednako za sve slojeve mreže. Sljedeći problem kod korištenja neuronskih mreža bila i degradacija gdje povećanjem dubine mreže, sposobnost treniranja modela pada. Neka od zanimljivih prepreka koje je ResNet morao riješiti je i problem performanse koji se veže na prethodno spomenuti problem da s većom dubinom mreže modeli postaju zasićeniji, a pogreške prilikom treniranja sve su češće i veće što umanjuje performanse rada neuronskih mreža. ResNet je na takve stvari morao utjecati svojom novom, naprednijom i boljom arhitekturom i logikom iza nje. Glavna okosnica arhitekture ResNet modela jesu rezidualni blokovi o kojima je već bilo riječi. Na slici u nastavku može se vidjeti presjek jednog takvog bloka. Glavna razlika je to što u klasičnoj neuronskoj mreži podatak ulazi u mrežu, zatim prolazi kroz niz konvolucijskih slojeva i na posljetku prolazi i kroz funkciju aktivacije. U rezidualnoj arhitekturi ResNet-a, podatak ulazi u mrežu, prolazi kroz konvolucijski sloj i onda se na izlazu iz konvolucijskog sloja, ulazni podatak opet dodaje prije primjene aktivacijske funkcije. Rezidualna mreža ovdje pomaže u očuvanju informacija iz originalnog ulaza dodavanjem tog ulaza izlazu konvolucijskog sloja, što može pomoći u izbjegavanju problema s degradacijom performansi kod dubokih mreža. Kada bi se to prikazalo formulama, ona bi glasila ovako: $\mathbf{H}(\mathbf{x}) = \mathbf{g}(\mathbf{x} + \mathbf{F}(\mathbf{x}))$.

\mathbf{H} – rezidualni blok

\mathbf{x} – ulazni podatak

\mathbf{F} – konvolucijski sloj

$\mathbf{g}(\mathbf{F}(\mathbf{x}))$ – funkcija aktivacije



Slika 5-1 Rezidualni blok ResNet arhitekture (Preuzeto 26.06.2024. s

<https://medium.com/@ibtadaazem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9>)

Sljedeća komponenta ResNet arhitekture je takozvana veza preskakanja. U rezidualnoj mreži, veza preskakanja omogućava da ulazni podatak zaobiđe (preskoči) konvolucijski sloj i izravno doda izlazu konvolucijskog sloja. Time se formira rezidualni blok. Ova komponenta omogućava mreži da direktno prenese informaciju s početka bloka na kraj bloka, što može pomoći u očuvanju originalnih informacija i lakšem učenju dubokih mreža. Veoma je korisna kod vrlo dubokih mreža, gdje rješava spomenuti problem gubitka gradijenta i degradacije performansi koji su do tad predstavljali neke izazove s kojima su se duboke neuronske mreže suočavale. Iz ovoga je jasno da svaki rezidualni blok onda sadrži dva dijela bloka, a to su konvolucijski sloj mreže i veza preskakanja koja ih zaobilazi. Sama arhitektura temelji se na tome da se ti rezidualni blokovi slažu više njih zajedno, a postoje verzije ResNeta s čak 152 sloja rezidualnih blokova. Tako danas postoji nekoliko verzija: ResNet-34, ResNet-50, ResNet-101 i ResNet-152. Još jedna veoma bitna komponenta arhitekture je GAP tehnika (globalno prosječno udruživanje – eng. *Global Average Pooling*) [39].

Ova tehnika popularna je i u drugim metodama, a služi za izračun prosječne vrijednosti svake značajke iz prethodnog konvolucijskog sloja prije ulaska u novi. Značajke su pak rezultati koje je generirao konvolucijski sloj prilikom obrade slike poput tona, teksture, rubova u slici i slično. Svaka ta značajka vezana je na jedan od onih filtera koji se postavljaju na sliku koja ulazi u konvoluciju odnosno u proces obrade. Prednost korištenja GAP tehnike je to što ne sadrži parametre za optimizaciju prilikom treniranja. Možda to zvuči neobično budući da ne koristi optimizaciju, ali na taj način zapravo ubrzava cijeli proces, smanjuje potrebne resurse za obradu i

rasterećuje model, a podaci će ionako biti optimizirani prolaskom kroz ostale dijelove arhitekture. GAP slojevi kompatibilni su s konvolucijskim te se međusobno nadopunjuju kroz značajke koje generiraju. Zbog svih navedenih prednosti, ResNet arhitektura funkcionira na zanimljiv i dobro provjeren način te je zato i doživjela veliki uspjeh [40].

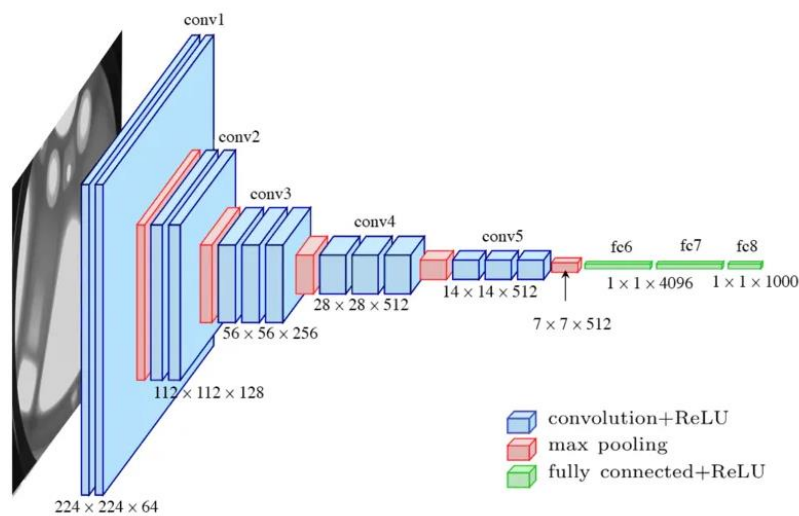
ResNet je svoju primjenu pronašao u raznim područjima s obzirom na lakoću korištenja, otvoreni kod dostupan svima i dobre rezultate. Prilikom obrade fotografija, ova se metoda koristi u mnogim aspektima, od detekcije objekata na svakodnevnim fotografijama, do analize satelitskih i medicinskih snimaka. ResNet se lako integrira s raznim okvirima (eng. *frameworks*) za detekciju gdje značajno poboljšava točnost u detekciji i klasifikaciji objekata unutar slike te omogućuje preciznije prepoznavanje i lokalizaciju objekata zahvaljujući kvalitetnijem izdavanju značajki. ResNet se koristi i prilikom obrade video podataka jer zbog svoje kompleksne, a opet jednostavne arhitekture omogućuje analizu slijeda slika, prepoznavanje obrazaca tijekom vremena i razumijevanje pokreta unutar videozapisa. Koristi se u sigurnosnim sustavima za prepoznavanje i praćenje sumnjivih aktivnosti, detekciju lica i sumnjivih objekata te praćenje pokreta. Također se koristi i u zabavnoj industriji i analizi videozapisa u raznim situacijama kao što su velika sportska događanja, koncerti, predstave i slično. Koristi se i u medicinske svrhe, ponajviše za analiziranje medicinskih snimaka. Često se koristi prilikom analiziranja CT ili MRI snimki i detekcije mogućih bolesti, anomalija i sličnih pojava na snimkama. ResNet i slični modeli dubokog učenja ključni su u medicinskoj dijagnostici zbog svoje sposobnosti detaljne analize slika, prepoznavanja uzoraka i pomoći koju pružaju liječnicima prilikom dijagnosticiranja bolesti i planiranja oporavka [41].

5.1.2. VGG

VGG je popularna metoda dubokog učenja za klasifikaciju i lokalizaciju objekata koja je razvijena na Oxfordskom sveučilištu, a trenutno postoje verzije VGG-16 i VGG-19 gdje prva verzija koristi 16 konvolucijskih slojeva neuronskih mreža, dok druga verzija koristi njih 19.

VGG također podržava preko 1000 različitih klasa za objekte u koje ih može svrstati kao primjerice lice, miš, tipkovnica, kuća, auto, motor i tako dalje. Arhitektura VGG modela grafički je prikazana na slici ispod teksta, a u nastavku je i nešto detaljnije objašnjena. Kao ulaznu fotografiju, ovaj model može prihvatiti sliku veličine i 224 x 224 u tri boje zelena, crvena i plava (RGB) što je vidljivo i iz Slike 4.12, a koju zatim kroz sljedeće korake obrađuje. Sljedeće komponente arhitekture odnose se na nekoliko konvolucijskih slojeva. Prva dva konvolucijska sloja koriste 64 filtera veličine 3x3, s ReLU aktivacijom. Sljedeća dva konvolucijska sloja

koriste 128 filtera veličine 3x3, također s ReLU aktivacijom. Svaka sljedeća tri sloja koriste 256 odnosno 512 filtera veličine 3x3, svaki s ReLU aktivacijom. Između slojeva, kao što je i na slici označeno crvenom bojom, postoje takozvani slojevi udruživanja (*engl. pooling*). Nakon svake grupe konvolucijskih slojeva, postoje spomenuti slojevi veličine 2x2 koji smanjuju dimenzije značajki kako bi se smanjila količina računalnih resursa potrebnih za obradu. Sljedeća i posljednja komponenta u arhitekturi jesu potpuno povezani slojevi koji koriste ReLU funkciju aktivacije (zeleno boja na slici: fc6, fc7 i fc8). Prva dva sloja fc6 i fc7 imaju 4096 neurona, a završni tj.izlazni fc8 sloj ima 1000 neurona što zapravo odgovara činjenica da se nad ulaznom slikom može provesti klasifikacija s 1000 klasa [42].



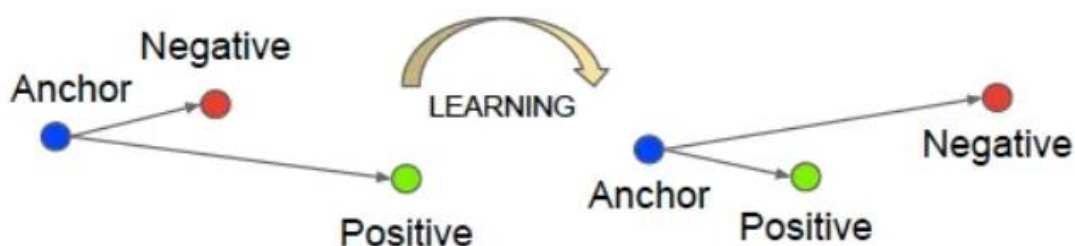
Slika 5-2 VGG arhitektura (Preuzeto 29.06.2024. s <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>)

6. Algoritmi za prepoznavanje lica

U ovom dijelu rada fokus će biti na algoritmima koji se najčešće koriste za prepoznavanje lica, a temelje se na dubokim neuronskim mrežama. Neće biti navedeni i objašnjeni svi algoritmi, već oni koji se u industriji najviše koriste.

6.1. FaceNet

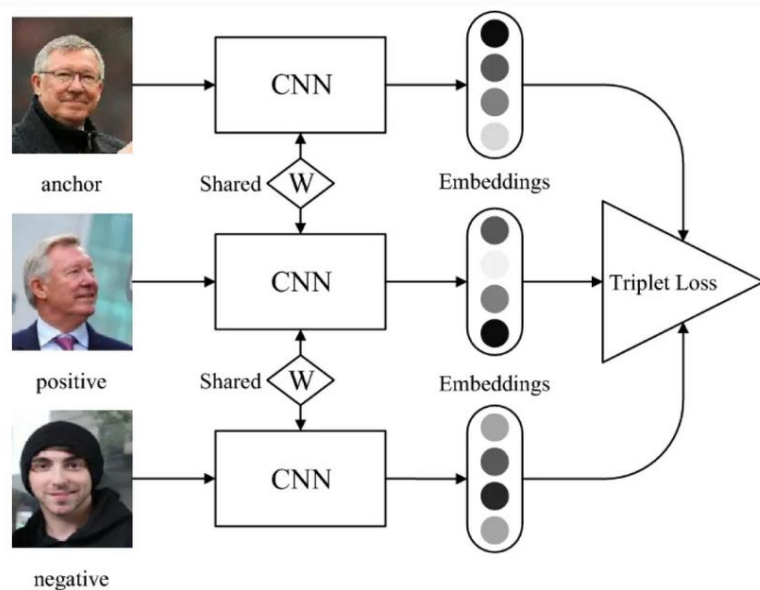
FaceNet je algoritam za detekciju objekata, a prvenstveno za detekciju lica kojeg su razvili Googleovi stručnjaci i inženjeri. Velika prednost algoritma zbog kojeg je odmah i doživio veliki interes i primjenu je mogućnost da iz slike izvuče njezine značajke na veoma visokoj razini kvalitete. Glavna tehnika na kojoj se FaceNet bazira naziva se trostruki gubitak (eng. *triplet loss*). Ova tehnika je zapravo način treniranja algoritama i modela da prepoznaju neke sličnosti i neke razlike između slika pomoću kojih se treniraju. Značajke slike nazivaju se trojke odnosno *triplet* zato što se dijele na tri grupe: glavne značajke na slici, značajke na slici koje su slične i značajke koje su različite. U YOLOv2 pojavio se pojam sidra koji zapravo označava unaprijed definirane okvire slike, a ova tehnika pokušava istrenirati modele da im sidra budu bliže onoj pozitivnoj od tri grupe koje se spominju u ovoj tehnici. Na slici je grafički prikazan način rada tehnike, prvo se sidro nalazi bliže negativnim značajkama odnosno različitostima na slici, a zatim se treniranje i učenjem uspješno postići da bude bliže pozitivnim odnosno značajkama koje su slične [43].



Slika 6-1 Triplet-loss tehnika (Preuzeto 28.06.2024. s <https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02>)

Na sljedećoj slici nalazi se praktičniji primjer ove tehnike za bolje razumijevanje. Kao što je moguće i primijetiti prve dvije osobe su iste ili veoma slične, dok je treća u potpunosti različita. *Triplet-loss* je u ovom slučaju odradila dobar posao i model je istreniran na ispravan način, tako

da su grupe značajki koje povezuju sličnosti postavljene što bliže, a slika koja je u potpunosti drugačija postavljena ima značajke što dalje od značajki sidrene slike. Značajke su vektori dimenzije 128.



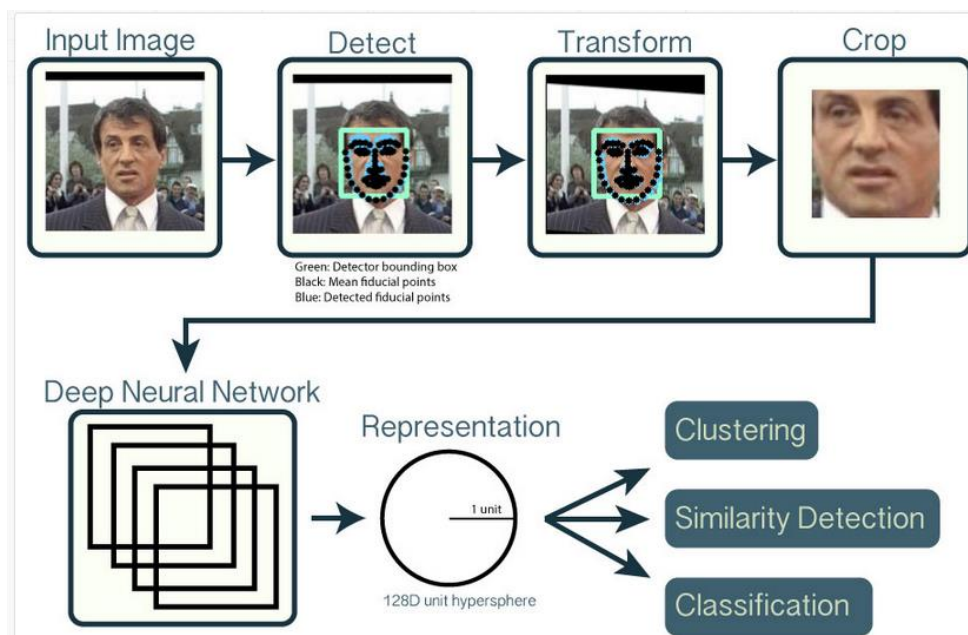
Slika 6-2 Triplet-loss praktični prikaz (Preuzeto 28.06.2024. s <https://www.v7labs.com/blog/triplet-loss#what-is-triplet-loss>)

FaceNet algoritam koristi se ovom tehnikom kako bi zapamtio značajke slika odnosno lica koja su stavljena blizu jedna drugih (pozitivna slika blizu sidrenoj slici na primjeru iznad) te kako bi ih kasnije u daljnjoj obradi mogao stavljati zajedno i tako tvoriti klaster odnosno skupove slika sa sličnim značajkama. Najčešća primjena algoritma je prilikom prepoznavanja lica gdje je pokazao zavidnu uspješnost, zatim kod identifikacije lica i kod stvaranja klastera s licima koja su obrađena uz pomoć FaceNet-a [44].

OpenFace je alat otvorenog koda koji se temelji na spomenutom FaceNet algoritmu, a također sadrži alate za detekciju lica, analizu ponašanja lica i prepoznavanje lica. S radom je započeo 2018. godine, a i dalje je prvi besplatni alat koji omogućuje detektirati značajke lica poput razmaka između očiju, pozicije glave i slično. Također je zanimljivo da je kompatibilan s mnogim aplikacijama te ga je veoma lako ugraditi u postojeće aplikacije [45].

Na sljedećoj slici nalazi se primjer rada OpenFace-a. Ulazna slika prikazuje poznatog američkog glumca Silvestra Stalona, a slika je preuzeta iz otvorene baze slika LWH Dataset. Prvi korak je detekcija lica na slici, a zatim slijedi transformacija te slike tako da oči i usta budu na istoj poziciji na svim slikama. Zatim slijedi korak gdje se koriste duboke neuronske mreže, točnije FaceNet algoritam unutar njih. U ovom koraku koristi se i prethodno objašnjena tehnika *triplet-loss*. Nakon što su slične slike detektirane i približene jedne drugima, povezuju se u

klustere nad kojima se kasnije može provoditi klasifikacija odnosno dodavati razne klase poput "muškarac", "mlad", "star" i slično [46].



Slika 6-3 OpenFace primjer korištenja (Preuzeto 28.06.2024. s <https://cmusatyalab.github.io/openface/>)

6.2. DeepFace

Sljedeći algoritam koji se često koristi razvijen je 2014. godine od strane Facebooka. DeepFace je tada donio revoluciju u prepoznavanju lica, ali je sa sobom i povlačio moralno pitanje s obzirom na to da su korištene tisuće slika korisnika ove društvene mreže. Iskorak nad drugim algoritmima bio je vidljiv i u tome što se DeepFace sada nije koristio samo za detekciju, već i predikciju s obzirom na to da je naglasak više na klasifikaciji slika. Jednostavno rečeno, kada je algoritam prepoznao okvire unutar kojih je žensko lice naprimjer, više fokus nije bio samo na tome da se identificirana detektirana osoba, već joj se pridodaju i klase poput "ljubazna", "vjernik", "stroga" i drugi epiteti koji su dodani putem predikcije o tome kakva bi ta osoba sa slike mogla uistinu biti [47].

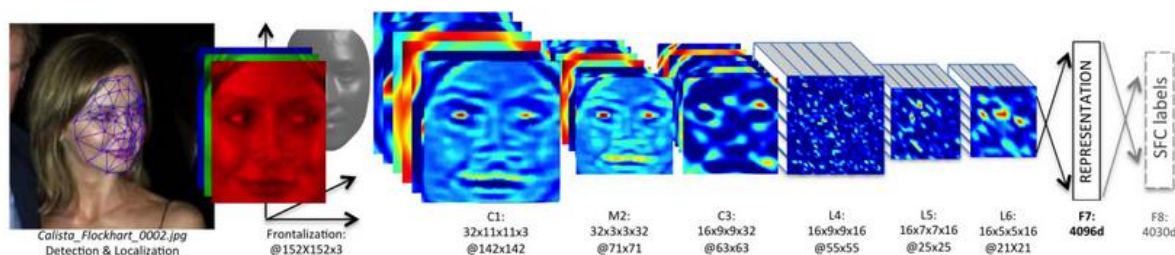
DeepFace je prvi algoritam koji se dosta oslanja na umjetnu inteligenciju, a veoma je jednostavan za korištenje i dostupan svima. Instalira se kao dopuna Pythonu te se pokreće brzo i lako. Ovaj je algoritam moguće koristiti za verifikaciju detektiranih lica što znači da se dvije ili više slika međusobno uspoređuju prema određenim značajkama te je moguće izvesti zaključak radi li se o istim ili različitim osobama. Koristi se također za detaljnu analizu tih značajki koje

predstavljaju fizičke atribute na licu kao što su razmak između očiju, pozicija očiju, usta i nosa, veličina i oblik čeljusti te površine čela. Velika je prednost algoritma što ga je moguće koristiti u stvarnom vremenu i to čak i na video zapisima, kamerama i slično.

DeepFace je pod vlasništvom američkog sveučilišta MIT, a zapravo je integriran u programski jezik Python te ga je moguće skinuti kao dodatnu instalaciju (PyPI) ovom jeziku te se može koristiti u privatne i poslovne svrhe [48].

DeepFace algoritam obrađuje slike u dva koraka: ispravljanje kuta i korak dubokog učenja. U prvom koraku pokušava se riješiti problem ako na slici osoba ne gleda direktno u kameru odnosno ako je lice okrenuto u nekom nepreglednom smjeru. To se ispravlja uz pomoć 3D modela koji "okreće" lice prema kameri. Nakon što je slika ispravljena i lice pravilno pozicionirano, procesiranje ulazi u sljedeću fazu, a to je korištenje dubokog učenja, točnije raznih modela dubokog učenja za prepoznavanje objekata/lica. Zanimljiva je činjenica da će DeepFace nakon analiziranja slike, ako generira veoma slične značajke za obje slike, ukoliko se koristi sijamska mreža, kao izlazni rezultat prikazati obje, umjesto samo jedne kao većina algoritama [49].

Na sljedećoj slici nalazi se praktični prikaz arhitekture DeepFace algoritma.



Slika 6-4 DeepFace arhitektura (Preuzeto 29.06.2024. s

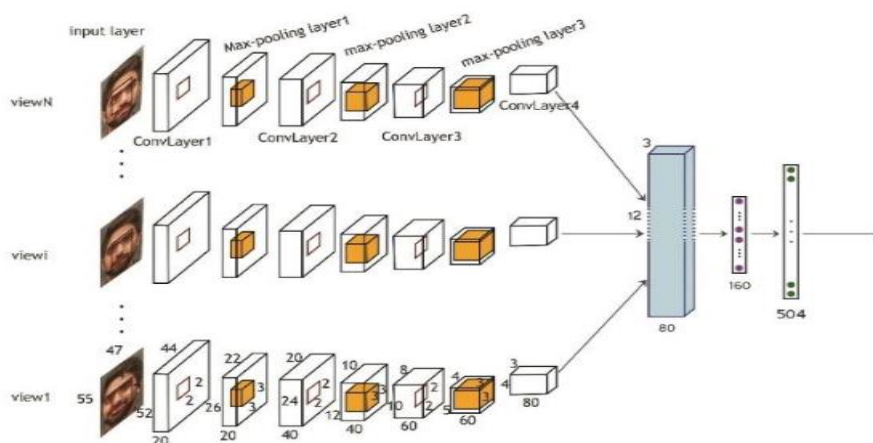
<https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>)

Prvi korak je detekcija lica, zatim slijedi faza ispravljanja kuta te se lice prilagođava da gleda prema kameri uz pomoć 3D modela, odnosno da su sve značajke lica jasno vidljive. Na prvom sloju C1 odvija se konvolucijsko filtriranje odnosno generiraju se značajke slike veličine 142 x 142. Sljedeći je M2 sloj koji generira značajke veličine 71 x 71 i dodatno obrađuje i filtrira značajke prethodnog sloja. Svaki sljedeći sloj radi ekstrakciju nekih značajki te smanjuje veličinu istih sve dok se u posljednjem sloju ne radi o ekstrakciji značajki 16 x 16. Nakon toga slijedi reprezentacija podataka (vektor dimenzije 4096) koja zapravo služi za klasifikaciju istih, povezivanje klasa sa slikama u bazi i slično. U posljednjem sloju arhitekture se zapravo izlazi iz neuronske mreže u kojoj je algoritam te se donosi finalni zaključak o identifikaciji lica [25].

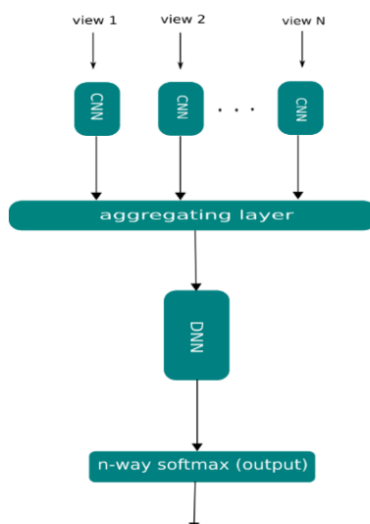
6.3. DeepID i M² Deep-ID

Nakon DeepFace-a, sljedeći algoritam koji je također odigrao veliku ulogu u detekciji i identifikacija lica je DeepID razvijen 2014. godine, čiji su autori kineski znanstvenici sa sveučilišta u Hong Kongu na čelu s profesorom Xiaoou Tang [50].

Novost u M² Deep-ID algoritmu, baziranom na DeepID, koju ovaj algoritam donosi sa sobom je arhitektura koja na ulazu koristi više pogleda (engl. *multiview*). Arhitektura s više pogleda omogućuje da se lice detektira iz različitih kutova (pognute, izdignute glave, iz lijevog i desnog kuta i slično), a to se postiže tako da se prilikom treniranja ovakvih modela koriste iste slike, ali iz različitih kutova. DeepID se fokusira na pet točki lica: dvije središnje točke očiju, vrh nosa i dva kuta usana. Na slikama u nastavku nalazi se arhitektura ovog modela odnosno algoritma.



Slika 6-5 DeepID arhitektura (Preuzeto 29.06.2024. s <https://arxiv.org/pdf/2001.07871>)



Slika 6-6 DeepID arhitektura (Preuzeto 29.06.2024. s <https://arxiv.org/pdf/2001.07871>)

Arhitektura se sastoji od nekih četiri glavna konvolucijska sloja koji također sadrže svoje pojedinačne komponente. Prvi dio arhitekture odnosi se na tehniku više pogleda. DeepID koristi duboku neuronsku mrežu za prepoznavanje lica koristeći 2D slike kao ulazne podatke. Ovaj modul je sastavljen od nekoliko pod-mreža, od kojih je svaka specijalizirana za određeni raspon pogleda (kutova lica). Ti rasponi su sljedeći (visina x širina x dubina): prvi sloj u rasponu 26 x 22 x 20, drugi sloj raspona 12 x 10 x 40, treći sloj je raspona 5 x 4 x 60, a posljednji ima raspon 4 x 3 x 80. Sljedeći sloj u arhitekturi naziva se agregacijski sloj. U prethodnom sloju sva se ova četiri raspona mogu smatrati pod-slojevima mreže, a pri izlasku iz mreže oni se svi skupljaju u jedan reprezentativni agregacijski sloj. Taj sloj zapravo povezuje sve poglede (kutove) lica i povezuje ih u cjelinu. Može se i prikazati formulom $\mathbf{d} = (\mathbf{N} \times \mathbf{H}) \times \mathbf{W} \times \mathbf{D}$, gdje N označava broj pogleda, H visinu zadnjeg sloja u prethodnoj mreži, W širinu, a D dubinu. Sljedeći sloj u arhitekturi je sloj učenja agregacijskog sloja gdje kombinacijom agregacijskog sloja i potpuno povezanih slojeva koje generira, model može bolje naučiti i prepoznati lica iz različitih pogleda. Može se reći da je to sloj treniranja modela. Posljednji sloj je *softmax* sloj koji generira krajnje rezultate obrade. Vektor značajki ima dimenziju 160 (kao i kod DeepID). Za bazu slika s 504 različite osobe, zadnji softmax sloj za klasifikaciju ima 504 izlaza kako bi se donio zaključak o identifikaciji osobe na slici. Osim softmax sloja, u ostalima se koristi ReLU aktivacijska funkcija [51].

Softmax aktivacijska funkcija pomaže da različite rezultate koje su dobivene kroz korake u radu modela, pretvorimo u lako razumljive postotke koji zbrajanjem daju 100%, omogućujući nam da vidimo koji je ishod najvjerojatniji. Recimo da je model trebao detektirati tri slike s

objektima kuća, auto, brod. Modeli ne generiraju rezultate u postotcima, već u običnim brojkama 2, 4, 1.5, a softmax funkcija to pretvara u razumljive i lako interpretirajuće podatke. U pravilu rezultate interpretira u rasponu između 0 i 1, gdje sve bliže nuli znači malo podudarnost, a sve bliže jedinici znači veću podudarnost. Ova se funkcija uvijek koristi na posljednjim slojevima neuronskih mreža, nakon što je slika procesirana do kraja [52].

6.4. VGGFace i VGGFace2

Nakon nekog vremena, kao i većina drugih modela i VGG model za prepoznavanje lica se razvijao te je u 2017. godini predstavljena novija verzija VGGFace2 temeljen na ResNet-50 ili SE-ResNet-50 arhitekturi neuronske mreže [trenutni-42]. Predložena je nova baza podataka tj. slika koja se primjenjuje u VGG modelu za prepoznavanje lica, stručni engleski naziv je *dataset* i bit će korišten u izvornom obliku u nastavku teksta. VGGFace2 baza sadrži preko 9 000 slika poznatih ličnosti s oko 3.1 milijuna raznih značajki tih lica. Pazilo se da u ovoj bazi bude svega, pa je tako puno slika s različitim osvjetljenjem, pozicijom ljudskog lica, omjer muškaraca i žena je skoro pa ravnopravan, prikazane su osobe raznih životnih dobi te je za svaku osobu prikazano oko osamdeset i nešto slika, a sve to kako bi se treniranje nad modelom izvršilo što točnije i preciznije. U nastavku se može vidjeti jedan primjer iz ove baze koji prikazuje slike princeze Haya Bint Al Hussein.



Slika 6-7 VGGFace2 primjer slika (Preuzeto 1.7.2024. s

<https://ar5iv.labs.arxiv.org/html/1710.08092>)

Da bi se slika u VGGFace2 modelu i bazi procesirala do kraja, postoje dva načina koja moraju biti zadovoljena, podudaranje lica u različitim pozama i podudaranje lica u različitim dobnim skupinama. U scenariju podudaranja lica u različitim pozama, za istu osobu mora postojati pet slika s konzistentnom pozom, a ta poza može biti slikano s profila, gledajući ravno u kameru ili da je vidljivo barem tri četvrtine lica. U scenariju podudaranja lica prema dobnim skupinama, baza se dijeli na one mlađe od 34 godine i starije. Kako bi se slike ispravno obradile

i prošle kroz metodu u ovoj bazi, postojalo je nekoliko koraka koji su objašnjeni ukratko u nastavku.

Prvi korak je odabrati neku javnu ličnost za koju se može garantirati da ima velik broj dostupnih slika.

1) Kako bi se dobile slike s velikim varijacijama u pozama i dobi, dodaju se ključne riječi poput obilježja oko dobne skupine (mlado, sredovječno, staro) te obilježja vezana uz pozu lica (frontalno, sa strane i slično). Na temelju toga, preuzima se još slika s novim obilježjima.

2) U ovoj se fazi zapravo koristi MTCNN za detekciju lica i VGGFace za izdvajanje značajki lica.

3) U prethodnoj fazi slike su se povezale s identitetom, zahvaljujući i dobro posloženim klasifikatorima. U ovom koraku cilj je maknuti klasifikatore s onih lica kojima ne pripadaju kroz razna treniranja i učenja modela kako bi se ponovnim prolaskom greške svele na minimum.

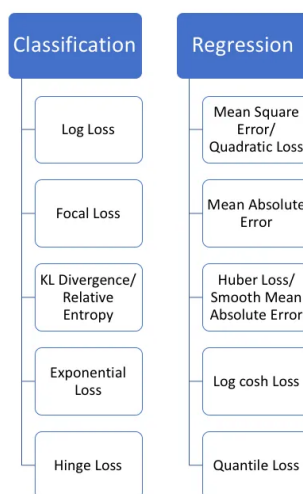
4) U ovom se koraku brišu duplikati slika.

5) Iako su se u prethodnim koracima izbrisale duple slike, klasifikatori su se pridružili pravim slikama, osobe su identificirane ispravno, uvijek postoji onaj ali i nekoliko izuzetaka koji nisu dobro procesirani. Ova faza koristi kombinaciju ručnih i automatiziranih algoritama za rješavanje ovih problema.

6) U fazama nakon ove ponovno se pokreće treniranje modela i učenje ako je potrebno. [53]

7. Funkcija gubitka u prepoznavanju lica

Duboke neuronske mreže itekako mogu biti podložne pogreškama i netočnim rezultatima. U tom slučaju funkcija gubitka može biti jako dobar pokazatelj što se s našim modelom događa i koliko je pogriješio u odnosu na procjenu. Koriste se kako bi pokazale razliku između rezultata koje je model predvidio i generirao te onih stvarnih rezultata koji su trebali biti generirani jer su točni. Uz pomoć funkcije gubitka mogu se pratiti performanse modela, predlagati poboljšanja, konstantno razvijati i trenirati model. Kratki hodogram i uvid u to kako radi ova funkcija može biti prikazan sljedećim koracima na primjeru predikcije cijene bicikla. Prvo se započinje sa treningom modela koji za trening koristi povijesne podatke o cijenama bicikala. Na temelju tih povijesnih podataka model radi predikciju nove cijene. Nakon toga dolazi do izračuna funkcije gubitka kako bi se izmjerila razlika između predviđene cijene i stvarne cijene bicikla. Unutar funkcije izračunava se i totalni gubitak, odnosno koliko je točno model pogriješio. Ukoliko je taj gubitak velik, parametri modela se ponovno definiraju s izmijenjenim vrijednostima u nadi da će ovaj put dati bolje rezultate te da će gubitak biti minimalan. Nakon što se ponovno pokrene proces treniranja potrebno je ponovno izračunati i funkciju gubitka. Ako je gubitak ponovno prevelik, parametri se iznova definiraju i model se trenira dokle je god gubitak u normalnim razmjerima odnosno dok predviđena cijena bicikla nije što točnija stvarnoj. Ovaj primjer slikovito opisuje rad algoritma unatrag (engl. *backpropagation algorithm*) prilikom treniranja neuronskih mreža. Funkcija gubitka može se podijeliti na dvije glavne vrste: funkcija gubitka regresije i funkcija gubitka klasifikacije, a na slici 7.1 može se vidjeti koje konkretne funkcije pripadaju ovim vrstama [54].



Slika 7-1 Podjela funkcije gubitka (Preuzeto 10.7.2024. s <https://heartbeat.comet.ml/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>)

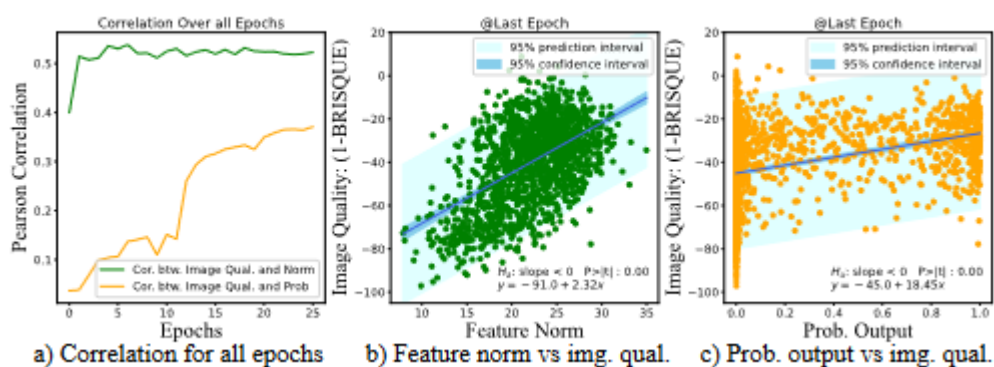
Algoritmi za regresiju predviđaju kakva će biti neka kontinuirana vrijednost na temelju one ulazne, a taj primjer je upravo prethodni s predviđanjem cijene bicikla na temelju ulaznih (povijesnih) podataka o cijeni. Algoritmi za klasifikaciju predviđaju mapiranje slika odnosno dodavanje klasa koje su zapravo kategorije po kojima se slike sistematiziraju i na temelju kojih se onda dohvaćaju značajke slike [55]. S obzirom na to da ima veliki broj funkcija gubitka koje bi se mogle opisati, u ovom radu će se fokusirati samo na one koje su usko vezane za prepoznavanje lica i poboljšanja treniranja takvih modela.

7.1. AdaFace

Ova funkcija gubitka nastala je kao odgovor na problem prepoznavanja lica na slikama slabije kvalitete kako ih modeli ne bi označili neprepoznatljivima, nego i njih uzeli u obzir prilikom treniranja. AdaFace svoju svrhu ispunjava kroz dvije dodatne funkcije, funkcije margina i kroz normalizaciju značajki.

Funkcije margina su neki podskup unutar funkcije gubitka, a prilagođavaju razmak između značajki slike koje pripadaju različitim klasama. Dvije takve funkcije koje tvore AdaFace jesu ArcFace i MagFace, a objašnjene su svaka zasebno u nastavku rada.

Norme značajki su zapravo vektori koji opisuju značajke neke slike, a modeli neuronskih mreža preoblikuju ih u numeričke vrijednosti kako bi mogli raditi s njima. Norma značajki i funkcija margine glavne su komponente AdaFace-a te ne mogu jedna bez druge i moraju se međusobno integrirati. Norme značajki veoma su bitan pokazatelj kada se kombiniraju s kvalitetom slike što je vidljivo iz sljedećih grafova.



Slika 7-2 Odnos normi značajki i kvalitete slike (Preuzeto 10.7.2024. s

https://openaccess.thecvf.com/content/CVPR2022/papers/Kim_AdaFace_Quality_Adaptive_Margin_for_Face_Recognition_CVPR_2022_paper.pdf)

Sva tri grafa prikazuju neke faze (epohe) treniranja modela. Na prvom grafu prikazan je odnos Pearsonove korelacije i kvalitete slike. Pearsonova korelacija je mjera koja pokazuje ovisnost dva broja međusobno. Ako se kreću zajedno u isto vrijeme i u istom smjeru tada su u pozitivnoj korelaciji, u suprotnom se nalaze u negativnoj korelaciji [56]. Pearsonova korelacija u pogledu prepoznavanja lica može pripomoći kod poboljšanja performansi modela, visoka korelacija može značiti da nam model radi ispravno tj. da uči one značajke koje su bitne prilikom ispravnog klasificiranja objekta. U AdaFace funkciji, Pearsonova korelacija je korištena za analizu povezanosti između norme značajki i kvalitete slike, kao i između izlazne vjerojatnosti modela i kvalitete slike. Prilikom treninga, Pearsonova korelacija između norme značajki i kvalitete (definirane preko nereferentne BRISQUE mjere kvalitete, tj. 1-BRISQUE) brzo dolazi na najveću razinu, što znači da se može koristiti kao informacija za kvalitetu slike od početka treniranja. To je prikazano i na prethodnom grafu u zelenoj boji. Na posljednja dva grafa (b i c) prikazana je i posljednja faza učenja modela. Vidljivo je da na grafu b postoji pozitivna korelacija što znači da postoji pozitivna korelacija između norme značajki i kvalitete slike. Rezultat grafa c pokazuje nelinearnu korelaciju, te se vidi da slike niske ili visoke kvalitete mogu biti "lakše" ili "teže" za prepoznavanje. "Lakši" i "teži" uzorci tj. slike su definirane su kao 1- vjerojatnost prepoznavanja. Baš je iz tog razloga razvijen AdaFace kako bi regulirao ovaj posljednji graf, odnosno smanjio mogućnosti da se lošije slike lica krivo detektiraju i naposljetku i krivo identificiraju ili da model uopće ne pokuša procesirati takve slike. AdaFace kod treniranja koristi slike: a) s većom kvalitetom (tj. većom normom) i tada više koristi "teže" uzorke, b) s manjom kvalitetom (tj. manjom normom) i tada više koristi "lakše" uzorke (čime se izbjegavaju slike koje se ne mogu koristiti za prepoznavanje).

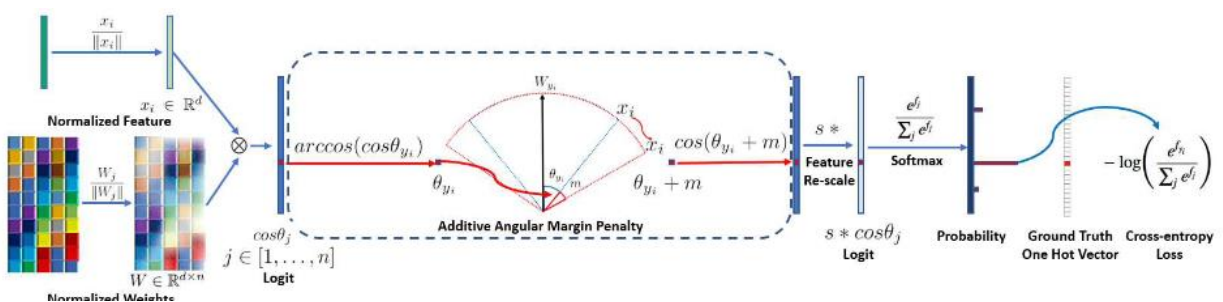
Kako bi AdaFace ispunio u potpunosti svoju svrhu, razvijena su i tri načina augmentacije koja bi mu pomogla. Augmentacija je poboljšavanje podataka u smislu povećanja, proširivanja tj. produbljivanja podataka. Kod treniranja modela za detekciju lica, augmentacija u pogledu dodavanja novih slika možda i nije toliko efektivna te se skupovi podataka često mogu povećati tako da se postojeće slike malo izmjene rotacijom, obrezivanjem i slično. AdaFace koristi se sljedećim augmentacijama: Izrezivanje (engl. *cropping*) gdje se definira jedno područje nasumičnim odabirom, a područje izvan njega postavlja se da je 0 odnosno da se ne koristi. Na taj način se slika ne reže doslovno, ali se s njome postupa kao da je taj dio izvan tog okvira odrezan i da kao da ne postoji. Sljedeća je fotometrijska augmentacija gdje se promjenom svjetline i zasićenosti nekih fotografija može proširiti skup podataka s novim fotografijama. Posljednja augmentacija odnosi se na promjenu veličine fotografija gdje se može smanjiti preglednost slike ako joj se smanji i veličina pa se na taj način model isto trenira kako raditi sa slikama slabe kvalitete [57].

AdaFace je nova i inovativna funkcija gubitka koja uz pomoć funkcije margina i norme značajki omogućuje detekciju i prepoznavanje niskokvalitetnih slika.

7.2. ArcFace

ArcFace je već ukratko predstavljen u prethodnom dijelu, a nešto detaljniji opis ove funkcije gubitka slijedi u nastavku.

Ova funkcija rješava razne probleme kod detekcije lica te sa sobom donosi novitete i prednosti. Pomoću ArcFace-a mogu se direktno optimizirati margine odnosno udaljenosti između značajki slike, također funkcija je veoma efektivna jer uspijeva odraditi posao nad velikim skupom podataka te čak i nad video zapisima i slikama odnosno licima koja se tamo pojavljuju. Prednost je i jednostavnost rada s ovom funkcijom s obzirom na to da je za njezinu implantaciju potrebno dodati svega par linija koda koje su već ionako javno dostupne svima u GitHub repozitoriju ArFace-a. Zanimljivo je da se može koristiti kao zasebna funkcija bez potrebe da se implementira s drugima te da će svejedno dati jednako dobre rezultate. Na sljedećoj slici grafički je prikazana implementacija ArcFace-a te je objašnjena u nastavku.



Slika 7-3 ArcFace implementacija (Preuzeto 10.7.2024. s

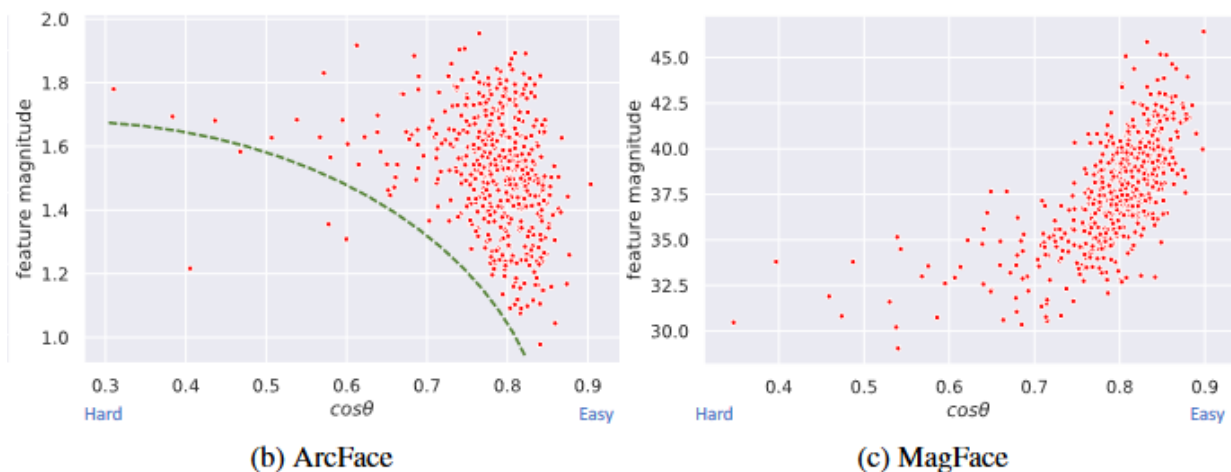
https://openaccess.thecvf.com/content_CVPR_2019/papers/Deng_ArcFace_Additive_Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.pdf)

Na slici iznad nalazi se jedan generalni proces treniranja modela temeljenog na neuronskoj mreži, ali uz dodatak da za funkciju gubitka koristi ArcFace. Prvo model prima ulazne značajke slike i normira ih u vektore odnosno numeričke vrijednosti kako bi s njima mogao raditi. Zatim slijedi nekoliko koraka specifični baš za ArcFace. Prvo se izračunava kosinus kuta. Kut u ovom slučaju označava kut između dva vektora, a koristi se da se usporede i procjene sličnosti između ulazne slike i vektora koji predstavljaju određene klase. Sljedeći korak je dodavanje dodatnog kuta na postojeći (na slici *Additive Angular Margin Penalty* dio) kako bi se udaljenost među

značajkama još malo povećala i na taj način smanjila mogućnost da se značajke preklapaju i da se lice krivo ili pogrešno identificira. Zatim se primjenjuje softmax funkcija u sljedećem koraku koja je već u nekoliko navrata objašnjena u radu. U posljednjoj fazi (na slici *Cross Entropy Loss*) računa se funkcija gubitka kao križna entropija u vrijednosti između onih predviđenih i onih stvarnih. Cijela poanta rada ArcFace-a temelji se na izračunu kutova i dodavanju novih između vektora tj. klasa kako bi se smanjilo preklapanje i kako bi se poboljšala točnost klasifikacije. Dodavanjem ovog margina (kuta), ArcFace omogućuje modelu da bolje razdvaja lica različitih identiteta, što dovodi do preciznijih rezultata u zadacima prepoznavanja lica te što manje razlike u predviđenim i generiranim rezultatima [58].

7.3. MagFace

Uz ArcFace i AdaFace, bitno je spomenuti još jednu relativno novu, ali naprednu funkciju gubitka, a to je MagFace. Prvi put do sada, pri prepoznavanju lica i radu modela u obzir se uzimaju i magnitude značajki prije nego se započne s normizacijom kako bi se dobile što kvalitetnije procesirane slike. Magnituda značajki je pojam koji se definira kao veličina vektora, a vektori su numeričke vrijednosti značajki slike. Ako je magnituda vektora (veličina) veća to može značiti da je i sama slika kvalitetnija jer ima veće tj. kvalitetnije značajke. MagFace funkcija zadržala je neke karakteristike od ArcFace pa tako svojim značajkama dodaje kutne margine kao i ArcFace. Uz to omogućuje da model ne postane prezasićen brojem značajki koje prima, a strukturirana je tako i da radi na klasterizaciji značajki odnosno da se u pravilu slike sistematiziraju i organiziraju u odgovarajuće grupe prema nekim sličnostima koje dijele. MagFace također koristi i korištenje izračuna kosinusa među tim kutovima koje je predstavljenu u prethodnoj funkciji. S obzirom na to da su ArcFace i MagFace toliko povezane, slika 7.4 prikazuje usporedbu te dvije funkcije.



Slika 7-4 Usporedba ArcFace i MagFace funkcije (Pruzeto 10.7.2024. s

https://openaccess.thecvf.com/content/CVPR2021/papers/Meng_MagFace_A_Universal_Representation_for_Face_Recognition_and_Quality_Assessment_CVPR_2021_paper.pdf)

MagFace je napredniji u odnosu na ArcFace jer koristi izračun kuta vektora (i to još adaptivni odnosno prilagodljivi svakoj značajki) te magnituda značajki, čime se poboljšava performanse rada modela, dok ArcFace koristi samo izračun margina kutova i to samo fiksni način što može dovesti do netočnih značajki. MagFace omogućava bolju obradu značajki, smanjuje i izbjegava šumove na slici i može još bolje obraditi slike niske kvalitete. Na oba grafa crvene točke predstavljaju jedan uzorak iz skupa slika nad kojima se model trenira. Os x predstavlja kosinus kuta, a os y vrijednost magnituda. S obzirom na to da ArcFace koristi fiksno područje unutar kojeg su značajke prihvatljive, vidljivo je sa slike da je dosta njih na rubnom dijelu tog područja ili izvan njega (zelena krivulja na slici). S druge strane MagFace nema takvo ograničenje te se sve značajke i kutovi prilagođavaju prema magnitudi značajke tj. veća im je ovisnost o y-osi što se vidi i s grafa. U gornjem desnom kutu MagFace grafa nalaze se najkvalitetniji uzorci jer je tamo korelacija magnituda i kosinusa kuta najjača.

Na kraju, može se zaključiti da svaka od ovih funkcija gubitka sa sobom donosi novitete i prednosti, no MagFace funkcija možda najbolje rješava problem obrade slika loše kvalitete jer uzima najbolje značajke iz svake funkcije te zajedno sa svojim vrijednostima magnituda pruža najbolje uvjete za rad modela [59].

8. Mjere učinkovitosti prilikom detekcije lica

U prethodnom dijelu objašnjene su mnoge metode, algoritmi i samim time i načini na koje je moguće detektirati i prepoznati osobe na slikama. Iako je cilj svakog treniranja modela da on da čim točnije i preciznije rezultate, 100%-tna točnost nikada ne može biti zagarantirana. Detekcija i prepoznavanje lica je tehnika koja se danas koristi u više-manje svim aspektima naših života, a u pojedinim situacijama može i spasiti ljudske živote, primjerice detekcijom video nadzornih kamera, sigurnosnih provjera na aerodromima i slično. Baš zbog svih navedenih, ali i mnogih drugih razloga, važno je da postoje mjere koje evaluiraju koliko je detekcija učinkovita kako bi se i modeli i algoritmi mogli dodatno razvijati i unaprjeđivati. Te se metode učinkovitosti sastoje od nekoliko ključnih stavki i raznih mjera, od kojih će neke biti dodatno pojašnjene.

Točnost je svakako prva i najvažnija mjera koja mjeri sveukupni broj točnih rezultata modela uspoređujući ih s netočnim rezultatima. Matematički gledano, točnost se izračunava dijeljenjem broja točnih i broja netočnih i točnih predviđanja. Kod detekcije lica, točni su samo oni rezultati u kojima je model uspio detektirati, ali i ispravno identificirati osobu. Sve ostalo su polovični podaci koji se ne mogu uzeti u razmatranje kao ispravni. Točnost sama po sebi nije dovoljna metrika za dubinsko analiziranje modela te se često koristi u kombinaciji s preciznošću i ostalim metrikama. Najveći izazov točnosti je dobro postavljena baza sa slikama, odnosno ako između klasa postoji veliki razmjer u broju slika koje joj pripadaju, rezultati će biti ne ispravni. Postoji rješenje i za to, a to je balansirana odnosno ukupna točnost. Kod takve točnosti rezultata je sredina između specifičnosti i odziva za pojedinu klasu. Odziv se iskazuje kao omjer između istinito pozitivnih i zbroja istinito pozitivnih i lažno negativnih rezultata.

Preciznost je veoma slična mjera kao prethodna te se često preciznost i točnost mogu zamijeniti. Kod točnosti se broje samo točne i ne točne vrijednosti, a kod preciznosti se prikazuje i broj lažnih pozitivnih vrijednosti. Može se reći da je preciznost broj točnih vrijednosti podijeljen zbrojem točnih vrijednosti plus lažnih točnih vrijednosti. Glavna razlika je ta što točnost ne mjeri te vrijednosti koje se prikazuju kao točne, ali zapravo nisu, dok preciznost više ulazi u dubinu rezultata i bolje ih filtrira. Kada se govori o detekciji lica, lažna točnost je onaj rezultat u kojem je model detektirao osobu, ali ju na poslijetku pogrešno identificirao. Kod točnosti je bitno da se osoba identificira, a kod preciznosti se provjera da li se je i točno identificirana, je li lice na slici povezano s ispravnim identitetom. Preciznost također ima svoje mane zbog čega se najčešće koristi u kombinaciji s drugim metrikama jer sama po sebi nije dovoljna. Mane ove mjere je što ne uzima u obzir pogreške sustava, kao primjerice nemogućnost da pronađe sve lažno pozitivne vrijednosti.

Osjetljivost ili stopa stvarno pozitivnih je mjera koja pokušava prikazati upravo ono što je i u nazivu same metrike, postotak stvarno točnih rezultata. Matematički gledano to bi značilo da se broj točno pozitivnih dijeli zbrojem točno pozitivnih i lažno negativnih. Iako lažno negativno zvuči kao da je to zapravo točno pozitivno, nije baš toliko jednostavno. U kontekstu osjetljivosti, točno pozitivan rezultata je onaj koji je točno identificirao osobu, a lažno negativni je onaj rezultat u kojem uopće nije uspješno pronađena nijedna osoba. Osjetljivost je metrika za koju je vrlo važno da ne zakaže u nadzornim kamerama i sigurnosnim sustavima.

Sljedeća važna i zanimljiva mjera je **specifičnost ili TNR** (skraćunica za engleski naziv *True Negative Rate*). Ova prikazuje i konkretan negativnih i točno negativnih rezultata. To je zapravo broj točno negativnih rezultata podijeljen zbrojem točno negativnih i lažno pozitivnih rezultata. Ova se mjera koristi na ulazu u države (sigurnosne kamere na granicama).

F1 mjera najviše se koristi u bazama slika koji nisu dobro balansirane (omjer muško žensko, mlado staro, dobar etnički omjer i slično). Ova mjera pazi i uzima u obzir sve negativne vrijednosti, i one točno negativne i one lažno negativne. Matematički gledano to je sredina između preciznosti i osjetljivosti, a računa se kao umnožak preciznosti i osjetljivosti podijeljen sa zbrojem preciznosti i osjetljivosti i sve se to množi dva puta. Kada je rezultat F1 mjerenja jednak broju 1, može se smatrati da je to najbolji rezultat. Kao i sve mjere, i F1 ima svoje mane, a jedne je svakako što u obzir ne uzima istinito negativne vrijednosti rezultata.

FAR i FRR su skraćenice za engleske nazive *False Acceptance Rate* i *False Rejection Rate* što u doslovnom prijevodu znači stopa lažno prihvaćenih i lažno odbijenih rezultata. Ove se mjere koriste zbog velike potrebe da modeli prate svoje greške i kako s njima postupaju odnosno da li ih prihvate ili odbiju, a sve u svrhu što preciznijeg i točnije treniranja modela i dobivanja najoptimalnijih rezultata. FAR je mjera koja ukazuje na to kolika je mogućnost da model identificira krivu osobu. Računa se tako da se podijeli broj lažnih identifikacija odnosno netočnih s brojem koliko je puta model pokušao napraviti identifikaciju. Ako model odnosno sustav koji koristi neki od modela ima visoku stopu FAR mjere, zapravo znači da je taj sustav dosta nepouzdan jer prima veliki broj lažno pozitivnih rezultata. FRR mjera je zapravo suprotno od FAR-a te izračunava stopu onih rezultata koji su zapravo točni, ali ih je model krivo predvidio i odbio ih te im dodao labelu da su netočni. Sasvim je razumno da kod FRR mjere želimo čim nižu stopu takvih slučajeva. Postoji i poveznica između te dvije mjere koja se naziva EER (eng. *Equal Error Rate*) što je zapravo točka gdje su te mjere izjednačene.

ROC (eng. *Receiver Operating Characteristic Curve*) je mjera koja omogućuje provođenje balansa između lažno pozitivnih i lažno negativnih rezultata. ROC krivulje ne koriste se toliko često kao neke druge mjere jer je realno veoma teško odrediti taj prag koji je prihvatljiv između lažno pozitivnih i lažno negativnih rezultata jer svakom sustavu odgovaraju drugačiji postotci.

GAR (eng. *Genuine Acceptance Rate*) je mjera koja prikazuje odnos ispravno identificiranih osoba na način da broj točno pozitivnih rezultata podijeli zbrojem točno pozitivni i netočno negativnih rezultata što u pravilu rezultira samo najtočnije rezultate. S ovom mjerom bi se u teoriji moglo definirati je li neki model odnosno sustav koji ga koristi skoro pa bez greške, no u praksi uvijek ima izuzetaka te nije preporučljivo slijepo vjerovati ovoj mjeri. GAR se često naziva i TMR odnosno *True Match Rate*.

Rank-N mjera se koristi u sustavima s unaprijed definiranom listom potencijalni ispravno identificiranih osoba. Ako je lista sadrži 20 kandidata (Rank-20), ispravni rezultat bio bi onaj koji bi identificirao osobu unutar prvih 20 podudaranja. Rank-N zapravo označava koja je vjerojatnost da se generira ispravna klasa u N najboljih predviđanja. Za što točnije rezultate, stopa mora biti što viša [60].

U praktičnom dijelu su neke od mjera učinkovitosti testirane te su prikazani rezultati testiranja i usporedbe između tih rezultata.

9. Praktični rad

9.1. Opis ideje

Motivacija za izradu ovog rada prvobitno je bila usmjerena na izradu web aplikacije i baze podataka s naglaskom na korištenje u domeni računalnog vida. Detekcija i identifikacija lica pokazala se kao dobar primjer koji bi mogao povezati sva tri područja, a isto tako dati i dovoljno teorijske potpore samom radu. Ideja je da se koristeći postojeću bazu slika (LFW) može identificirati osobe iz te baze. To je osmišljeno tako da se cijela zip datoteka sa svim slikama učita u aplikaciju, zatim se izdvoje značajke i klase slika, zatim se pohrane u bazu i na sljedećoj stranici aplikacije kada se učita samo jedna slika, iz baze se povuku značajke svih slika, trenira se model i koristeći taj isti model, osoba se identificira. Također je ideja da se isproba više algoritama, pa je tako moguće odabrati između FaceNeta i ResNeta. Sve je detaljnije objašnjeno u nastavku.

9.2. Korištene tehnologije

Za realizaciju ovog rada korištene su mnoge poznate i manje poznate tehnologije, programski jezici, alati i slično. Svaki od njih bit će ukratko objašnjen te će biti prikazana uloga koju ima u ovoj web aplikaciji.

SQLServer je tehnologija koja se koristi za manipulaciju podacima nad SQL relacijskim bazama. Relacijske baze jesu one u kojima su entiteti (tablice) povezani relacijama. Relacije se mogu smatrati kao neki odnosi između njih, najčešće su to 1:1, 1:N i N:M. U prvoj relaciji tablice ovise jedna o drugoj, odnosno jedan redak iz prve tablice povezan je samo s jednim retkom u drugoj. Najčešće su to redak koji je u prvoj tablici primani ključ tablice, a veže se na strani ključ druge tablice koji je zapravo redak koji povezuje obje tablice. U 1:N relaciji jedan redak tablice može biti povezana s više njih, a u N:M više redaka može biti povezano s više njih. U ovom radu nisu korišteni primarni i strani ključevi za povezivanje tablica zbog potrebe da se tablice s klasama uvijek brišu prije novog poziva skripte kako bi se upisali novi podaci, a kada tablice u SQL serveru koriste primarne i strane ključeve to nije moguće. SSMS je grafičko sučelje koje služi za povezivanje Microsoft SQL Servera s krajnjim korisnikom, a u ovom radu korištena je verzija servera odnosno SSMS-a 2020 i to Standard Edition zbog toga što sadrži sve značajke koje su dovoljne, a u besplatnoj su verziji.

Visual Studio Code poznati je editor u vlasništvu Microsofta za pisanje kodova. S obzirom na to da je dizajniran tako da podržava razne programske jezike, a isto tako ima i veliki broj

dodataka, bio je idealno rješenje za ovaj projekt s obzirom na to da je pomoću VS Code bilo moguće povezati puno različitih dijelova sustava u cjelinu.

Kao i kod svakog projekta i kod ovog je bilo veoma važno da se rad kontinuirano sprema, najviše zbog mogućih problema s laptopom, pa kako sav trud i rad ne bi bio izgubljen koristio se **BitBucket** koji je sustav poput GitHuba, a služi za pohranu raznih projekata te se izmjene spremaju na klasičan način, putem grana (engl. *brancheva*), zahtjeva za spajanje promjena (engl. *pull request*) i slično.

Od programskih jezika za realizaciju projekta korišteni su **Python, JavaScript, Node.js (Express), HTML te CSS**. Python je jezik koji je najviše korišten za rad s biometrijom te je tako bio i logičan izbor za ovaj projekt. Kroz godine se razvijao kao glavni jezik za rad dubokog i strojnog učenja te sadrži razne mogućnosti za obradu fotografija što ga čini idealnim za korištenje kod detekcije objekata na slikama. Danas je već toliko raširen u području biometrije da sadrži već mnoge dodatke, biblioteke i ostale pogodnosti za biometrijski rad. Node.js programski je jezik koji je zapravo neka nadogradnja JavaScripta i omogućuje da se JavaScript izvršava i izvan preglednika u kojem se koristi. U ovom je radu zapravo korištena podverzija Node.js koja se naziva Express, a služi kao web okvir za izradu web aplikacija. Za izradu ovog projekta bitan je jer zapravo se može definirati kao server koji služi za obradu upita (engl. *request*) i odgovora (engl. *respond*) koji se šalju kroz web aplikaciju. JavaScript je jezik koji najčešće korišten za izradu front-end dijela projekata odnosno za izradu dinamičkih web stranica. U ovom slučaju, web aplikacija pisana je u JavaScriptu, a za sam dizajn stranice korišten je popularni CSS jezik koji se i koristi samo za dizajniranje.

9.3. Arhitektura sustava

U nastavku je detaljnije prikazana cijela arhitektura i logika korištena u ovom projektu te potrebni preduvjeti kako bi se sama aplikacija mogla koristiti.

9.3.1. Preduvjeti za pokretanje i korištenje aplikacije

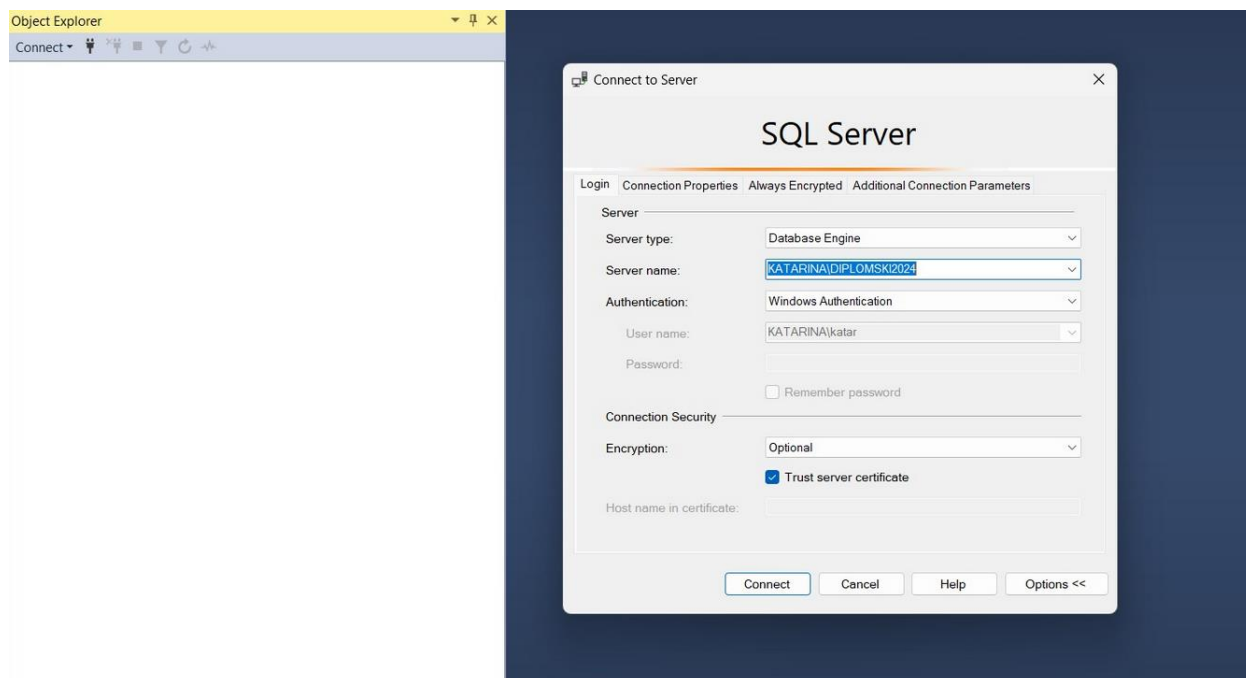
Za pokretanje aplikacije potrebno je podesiti nekoliko ključnih koraka, a nalaze se opisani u skripti README.txt u BitBucket repozitoriju.

Prvo što je potrebno instalirati ako korisnik nema na računalu je Visual Studio Code te SQL Server zajedno sa SSMS-om. Oboje su dostupni s verificiranih stranica tih alata. Kod instalacije SSMS-a bitno je u ovom slučaju odabrati da će se autentifikacija vršiti putem Windows autentifikacije, a ne SQL autentifikacije. Na taj način izbjegava se potreba za kreiranjem

korisničkog imena i lozinke prilikom svakog spajanja na server. Možda izgleda kao da nije najbolje sigurnosno rješenje, ali zapravo je potpuno suprotno. U slučaju da postoji lozinka i korisničko ime, veća je mogućnost probijanja te lozinke i ulaska u server. Kada je riječ o Windows autentifikaciji, serveru mogu pristupiti samo oni Windows korisnici koji su navedeni prilikom instalacije servera, a ako koristite Wizard za instalaciju, automatski će biti ponuđen korisnik koji je podešen na vašem osobnom računalu ili na virtualnom stroju ako se server i SSMS ne instaliraju lokalno. Nakon što je SQL Server instaliran i prilikom instalacije definirano željeno ime te instance servera, potrebno je kreirati bazu s odgovarajućim objektima kako bi aplikacija mogla negdje učítavati i od nekud čítati podatke. Za kreiranje navedenog najbolje je na samom serveru pokrenuti skriptu s BitBucket repozitorija "CreateDatabase.sql". Ako je jednostavnije, može se i preko grafičkog sučelja kreirati potrebna baza s objektima, a u nastavku su prikazani koraci na slikama.

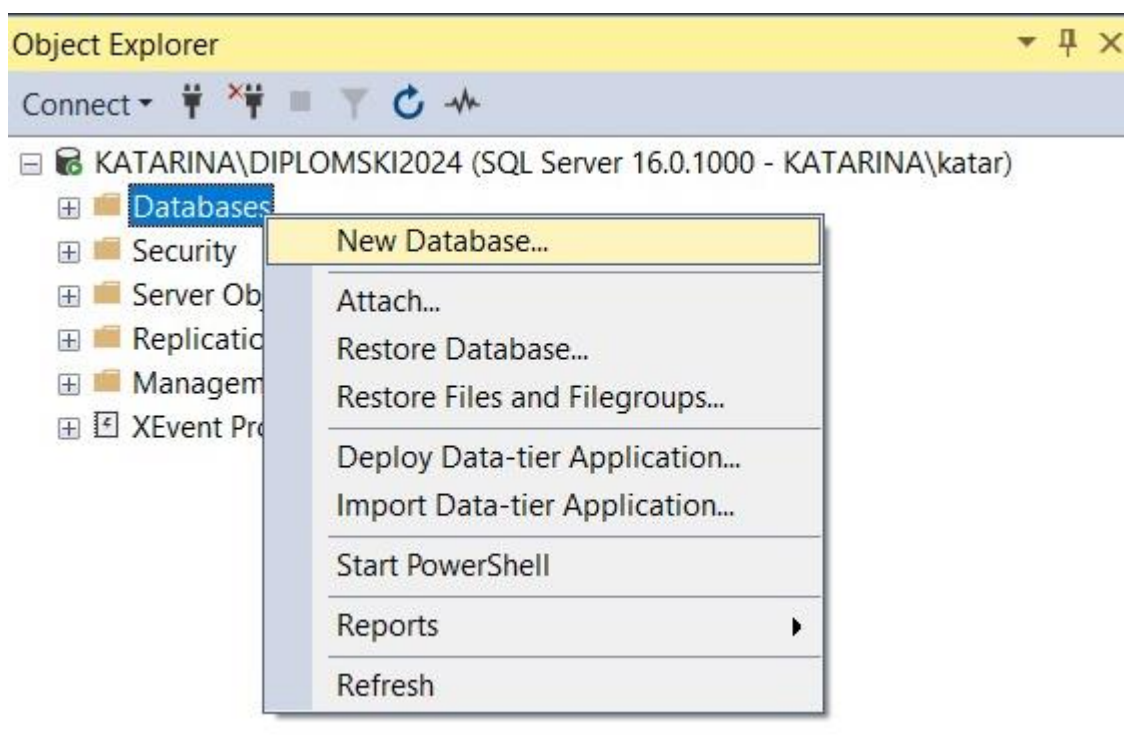
Nakon što je SQL Server instanca instalirana (moguće je koristiti upute s web stranice: <https://www.guru99.com/download-install-sql-server.html>), potrebno je u aplikacijama pronaći SSMS te ga pokrenuti. Prilikom prvog pokretanja, bitno je označiti polje "Trust Server Certificate" kako u budućnosti ne bi bilo poteškoća s povezivanjem aplikacijskih skripti sa serverom.

g



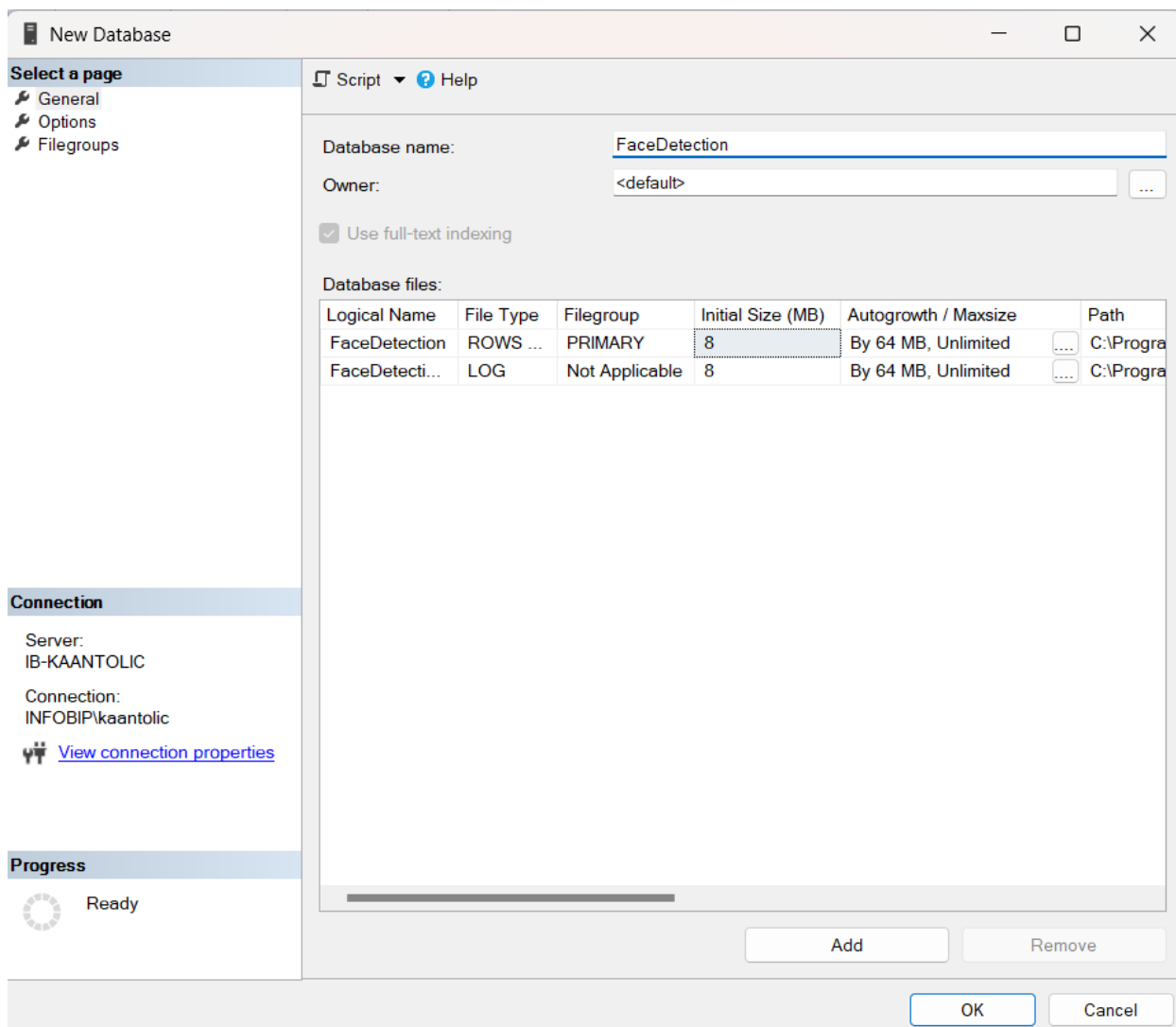
Slika 9-1 Spajanje na instancu SQL Servera (Vlastita izrada 10.08.2024.)

Nakon što je spajanje na server i instancu prošlo uspješno, potrebno je kreirati samu bazu odabirom "New Database" kao na slici ispod.



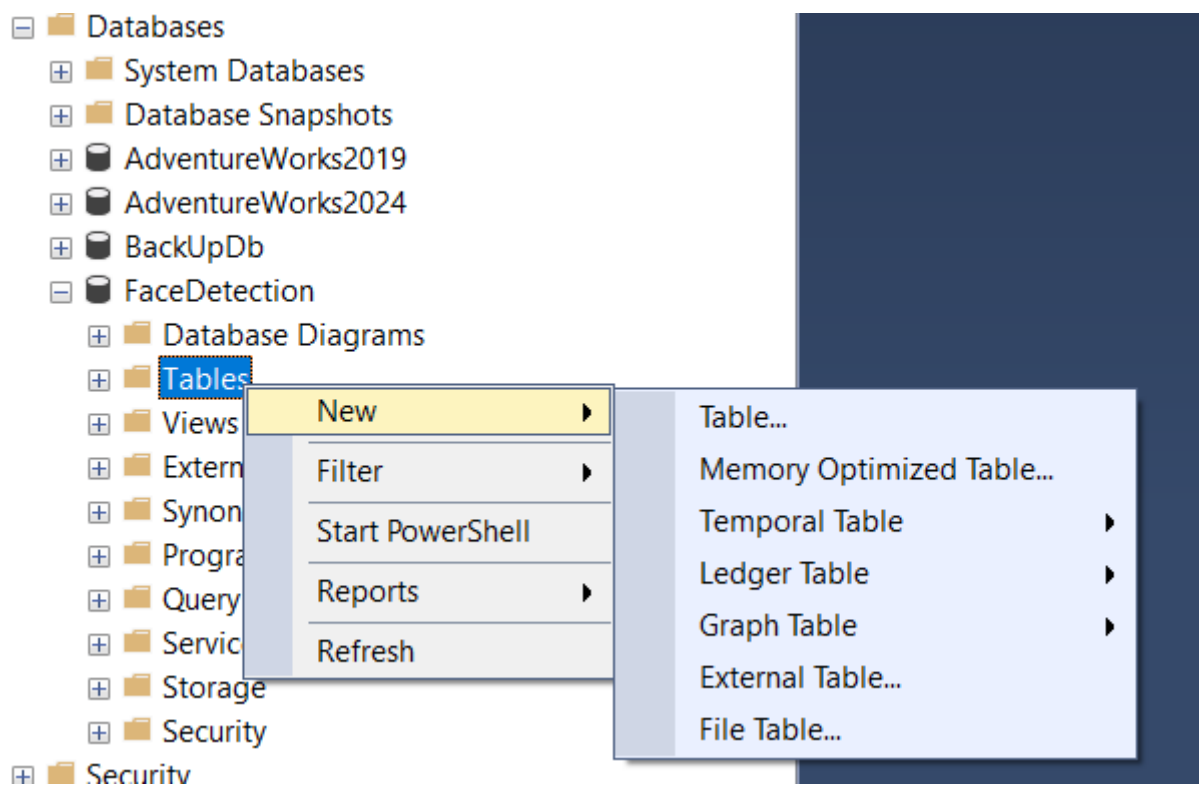
Slika 9-2 Kreiranje baze (Vlastita izrada 10.8.2024.)

Zatim se otvara novi prozor u kojem je u ovom trenutku jedino potrebno promijeniti ime baze, u ovom slučaju "FaceDetection". Sve ostale postavke, za korištenje izrađene aplikacije, dovoljno je ostaviti na zadanim postavkama te nije potrebno ništa dodatno definirati. Ako smo zadovoljni imenom, kliknemo gumb "OK".



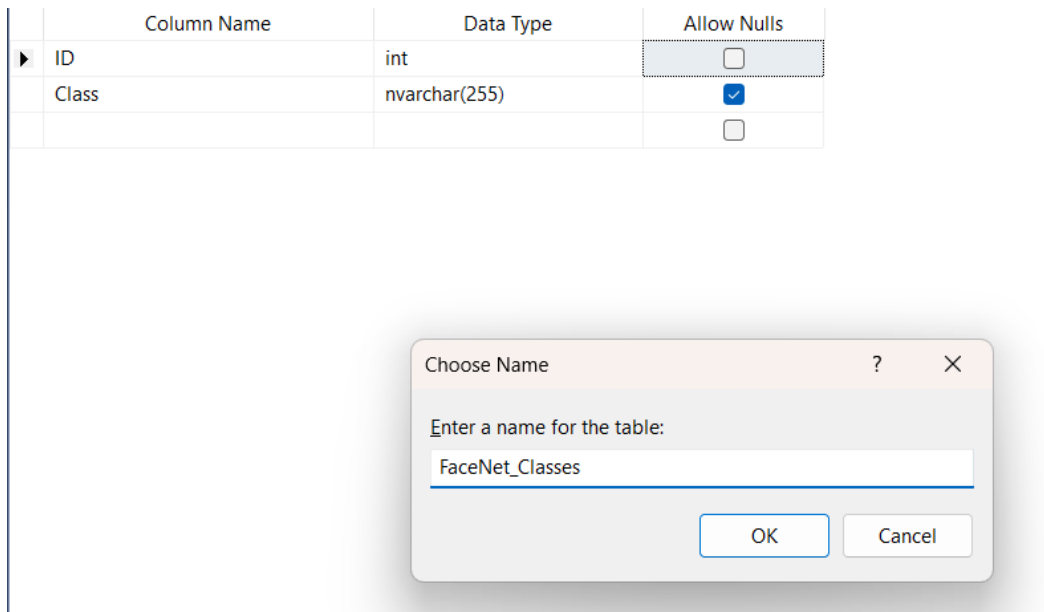
Slika 9-3 Imenovanje baze (Vlastita izrada 10.8.2024.)

Nakon što je baza kreirana, u lijevom stupcu možemo desnim klikom na "Databases" osvježiti listu baza i pronaći novo kreiranu bazu. Trenutno je baza prazna te je potrebno kreirati odgovarajuće objekte za ispravan rad aplikacije, u ovom slučaju četiri tablice, dvije za klase te dvije za značajke FaceNet i ResNet algoritama koji su korišteni u aplikaciji. Desnim klikom na znak plus ispred imena baze može se proširiti sadržaj baze te je potrebno lijevim klikom odabrati Tables – New Table kao na slici ispod teksta.



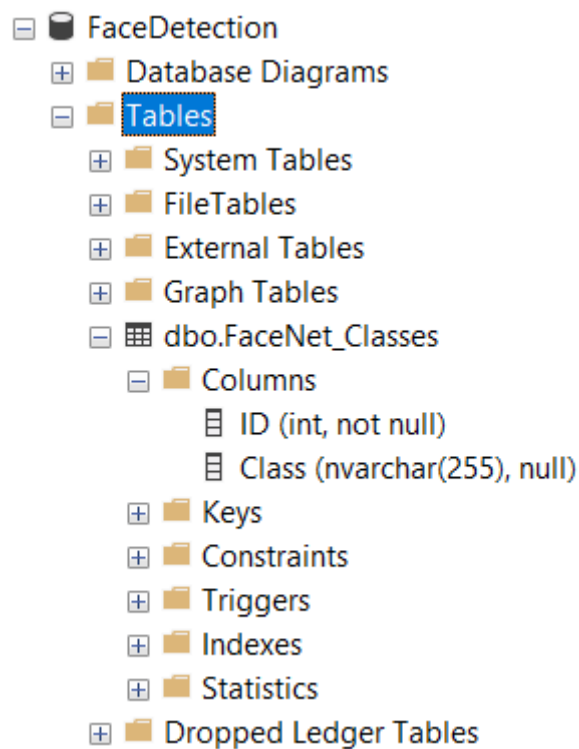
Slika 9-4 Kreiranje tablica (Vlastita izrada 10.8.2024.)

Slično i kao prilikom kreiranja nove baze, otvara se novi prozor u kojem će se definirati ime tablice i kolone koje sadrži. U našem slučaju za tablice FaceNet_Classes i ResNet_Classes potrebno je definirati dvije kolone: prva kolona imena ID, čiji tip podataka je int (skraćenica za *engl.integer* – brojevno) te je bitno u polju "Allow Nulls" za ID kolonu od značiti kvadratić s obzirom na to da se u ID kolonu ne smiju zapisivati prazni zapisi, odnosno uvijek mora biti popunjena. Nakon CTRL+S pojavljuje se skočni prozor u kojem je potrebno definirati ime tablice te odabirom "OK" tablica je kreirana.



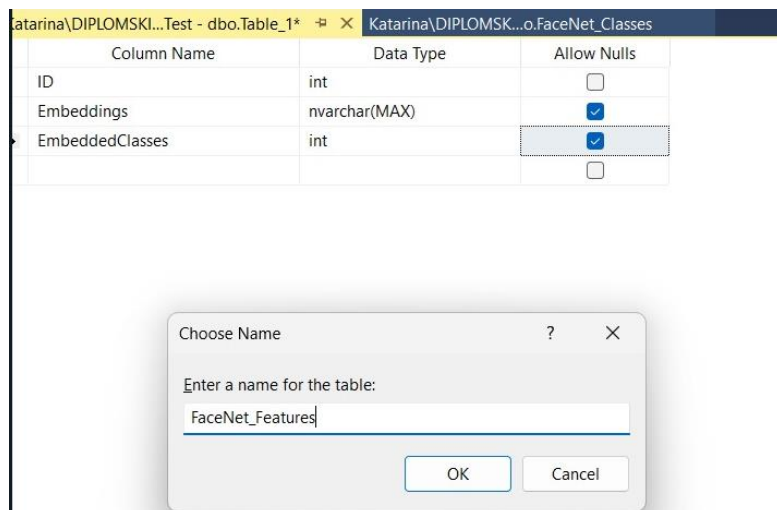
Slika 9-5 Imenovanje tablica (Vlastita izrada 10.8.2024.)

Posljednji korak je provjera u lijevom stupcu unutar SSMS-a da li se tablica uistinu kreirala i sa ispravnim kolonama.



Slika 9-6 Provjera kreiranih objekata (Vlastita izrada 10.8.2024.)

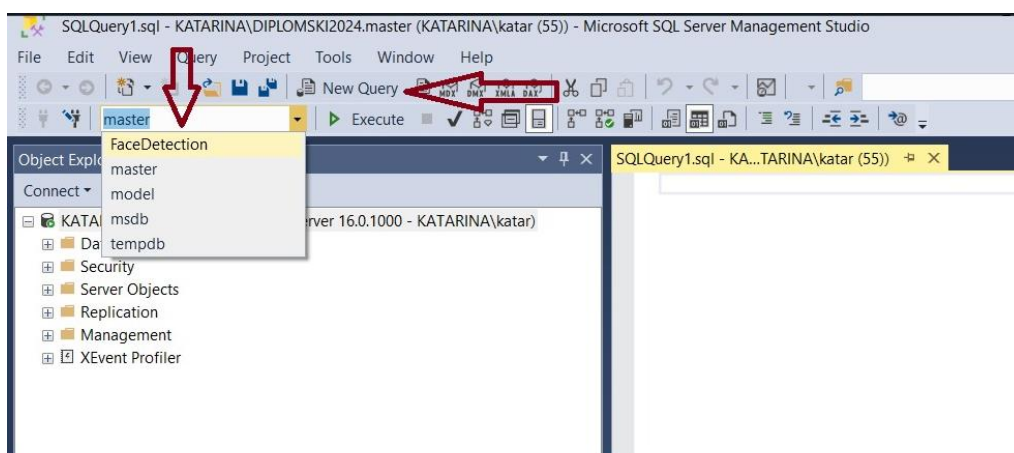
U prethodnim koracima prikazan je jedan primjer kreiranja tablice, isto je potrebno učiniti i za ResNet_Classes tablicu samo sa drugim imenom tablice. Kod kreiranja FaceNet_Features i Resnet_Features postupak je također isti, osim što se mijenjaju kolone i nazivi kao na slici ispod.



Slika 9-7 Kreiranje druge tablice (Vlastita izrada 10.8.2024.)

Nakon što su obje tablice kreirane, potrebno je još jednom provjeriti da li sada u bazi postoje sveukupno četiri tablice s određenim kolonama.

Ukoliko ništa od toga ne želite raditi ručno, potrebno je samo spojiti se na server i odabirom na "New Query" u alatnoj traci izvršiti skriptu "CreateDatabase.sql". Na slici je prikazano gdje se nalazi "New Query" opcija (ili koristite CTRL+N) te kako prilagoditi da se skripta izvrši na sistemskoj master bazi (iako se prilikom odabira "New Query", skripte automatski izvršavaju na toj bazi, dodatna provjera nije na odmet). Sistemska "master" baza koristi se prilikom kreiranja novih korisničkih baza. Kada se kreiraju tablice u FaceDetection bazi, potrebno je paziti da se uistinu i koristi navedena baza kako ne bi bile kreirane na pogrešnoj bazi, no navedena skripta podešava da se u svakom koraku koristi ispravna baza.



Slika 9-8 Odabir "New Query" opcije (Vlastita izrada 10.8.2024.)

Zatim kopirajte i zalijepite kod iz "CreateDatabase.sql" skripte te pokrenite gumb "Execute" ili F5. Skripta izvršava sve one korake iz prethodnog dijela, prvo kreira bazu, a zatim četiri tablice s pripadajućim kolonama.

```
SQLQuery1.sql - (lo...BIP\kaantolic (54))* X
CREATE DATABASE FaceDetection
GO

USE [FaceDetection]
GO

CREATE TABLE [dbo].[FaceNet_Classes](
  [ID] [int] IDENTITY(1,1) NOT NULL,
  [Class] [nvarchar](255) NULL
) ON [PRIMARY]
GO

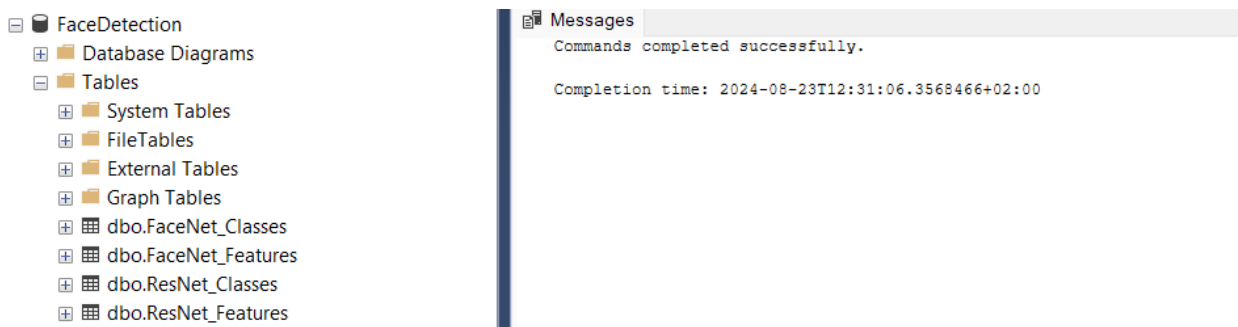
CREATE TABLE [dbo].[FaceNet_Features](
  [ID] [int] IDENTITY(1,1) NOT NULL,
  [Embeddings] [nvarchar](max) NULL,
  [EncodedClass] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[ResNet_Classes](
  [ID] [int] IDENTITY(1,1) NOT NULL,
  [Class] [nvarchar](255) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ResNet_Features](
  [ID] [int] IDENTITY(1,1) NOT NULL,
  [Embeddings] [nvarchar](max) NULL,
  [EncodedClass] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Slika 9-9 Pokretanje skripte za kreiranje baze i tablica (Vlastita izrada 10.8.2024.)

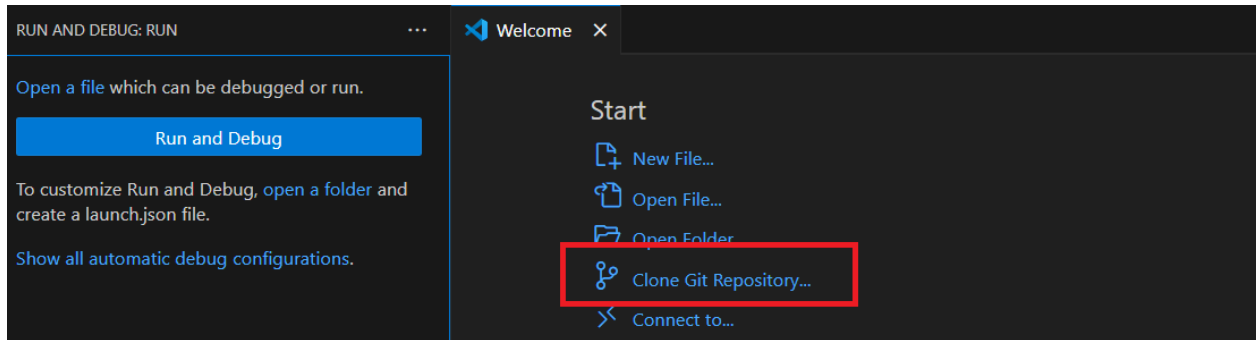
Ukoliko je sve prošlo kako treba, ispisuje se poruka "Command completed successfully", a u lijevom stupcu pojavit će se novokreirana baza s tablicama.



Slika 9-10 Provjera kreirane baze i tablica (Vlastita izrada 10.8.2024.)

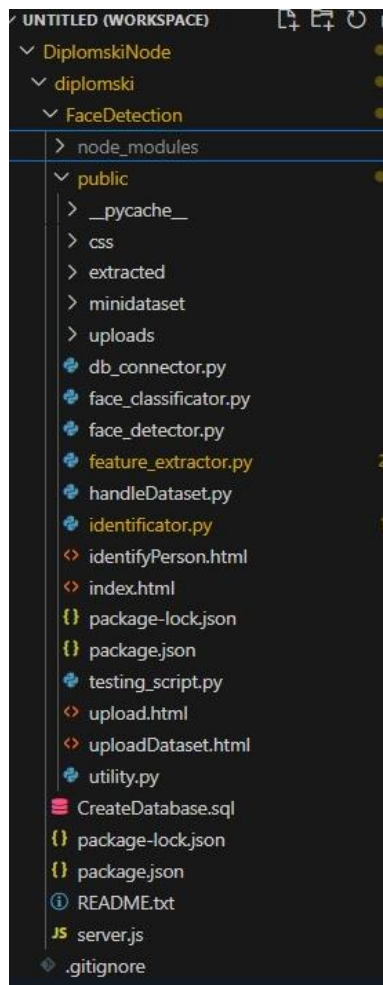
Nakon što je baza s pripadajućim objektima kreirana i ispravno podešena, potrebno je prilagoditi još nekoliko koraka. U prethodno instaliranom Visual Studio Code-u potrebno je klonirati cijeli BitBucket repozitorij, a ne ručno dodavati mape i skripte. Ako VS Code nema dodatne dodatke za GitHub, Python, JavaScript, HTML, JSON te CSS potrebno ih je instalirati.

Najprije je potrebno lokalno na računalu kreirati mapu u koju će se cijeli repozitorij klonirati, zatim u VS Code-u odabrati opciju prikazanu na slici ispod, a poveznica za kloniranje repozitorija koju je potrebno unijeti je sljedeća <https://diplomski2024-admin@bitbucket.org/diplomski2024/diplomski.git>.



Slika 9-11 Kloniranje repozitorija (Vlastita izrada 10.8.2024.)

Nakon što je repozitorij kloniran, struktura u Visual Code-u trebala bi biti kao na slici ispod.



Slika 9-12 Struktura kloniranog repozitorija (Vlastita izrada 11.8.2024.)

U ovom koraku, jedino što je još potrebno dodatno podesiti jest promijeniti zapisano ime SQL Servera u skripti db_connector.py. Na slici ispod teksta, nalazi se točan dio koda kojeg je potrebno izmijeniti. Ono što je potrebno izmijeniti jest ime servera sukladno onome koje je navedeno prilikom instalacije, te imena baze i tablica ako nisu ista. Ponekad će biti potrebno promijeniti i driver za baze, a to je moguće učiniti u ovoj skripti.

```
# Database constants
SERVER = 'KATARINA\DIPLOMSKI2024'
DATABASE = 'FaceDetection'
DRIVER = 'ODBC Driver 18 for SQL Server'
```

Slika 9-13 Promjena db.connector.py skripte (Vlastita izrada 11.8.2024.)

Prilikom instalacije Node.js unutar Visual Studio Code-a najbolje je pratiti standardne [upute](#). Node.js potreban je u ovoj aplikaciji zato što se koristi jedno njegovo okruženje Express koje zapravo predstavlja udaljeni server na kojem se odvija pozadinski promet u aplikaciji. Za instalaciju svih modula najbolje je koristiti npm (eng. *Node.js Package Manager*). U README.txt skripti također se nalaze sve potrebne pip instalacije koje je potrebno pokrenuti u terminalu nakon što se instalira. Nakon što je sve podešeno potrebno je pripremiti bazu slika na način da u jednu mapu postavimo više drugih mapa. Od toga svaka mapa predstavlja jednu osobu, a mora biti nazvana upravo ovom nomenklaturom: Ime_Prezime jer se kasnije naziv mape sa slikama osobe uzima kao klasa prema kojoj se detektira i identificira pojedinac. Primjerice, ako mapu sa slikama Jennifer Aniston nazovemo 'Test', ispod njezine slike ispisat će se da se zove 'Test'. Jednostavnije je koristiti postojeću bazu slika koju je moguće preuzeti [ovdje](#).

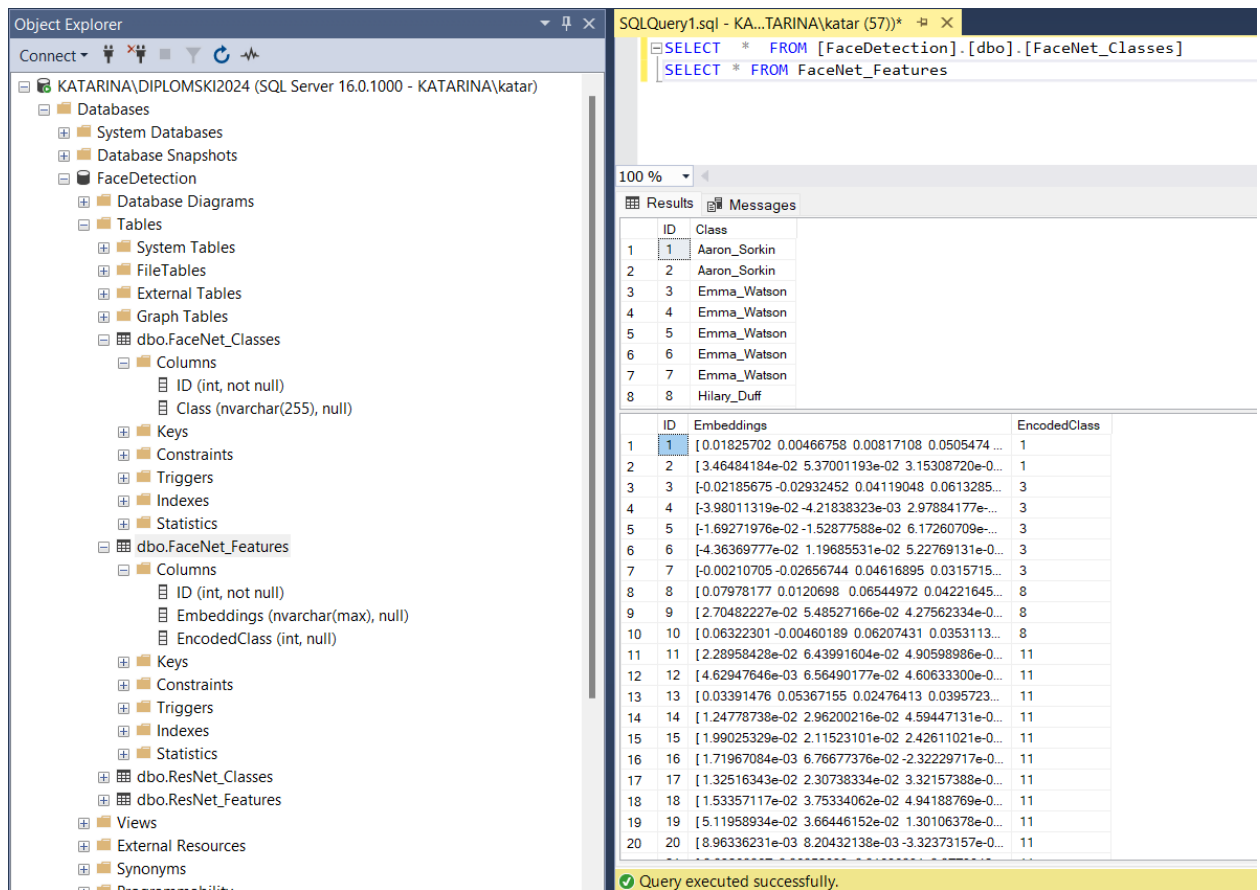
Također kako bi aplikacija radila, prvo je potrebno pokrenuti server na kojem se promet odvija. Potrebno je pozicionirati se u mapu FaceDetection unutar preuzetog repozitorija, a server se pokreće naredbom "node server.js" prikazano i na slici ispod teksta. Kada je sve podešeno ispravno, moguće je testirati aplikaciju na poveznici <http://localhost:3000>.

```
PS C:\DiplomskiNode\Diplomski> cd FaceDetection
PS C:\DiplomskiNode\Diplomski\FaceDetection> node server.js
Server is running on http://localhost:3000
```

Slika 9-14 Pokretanje servera (Vlastita izrada 25.08.2024.)

9.3.2. SQL Server

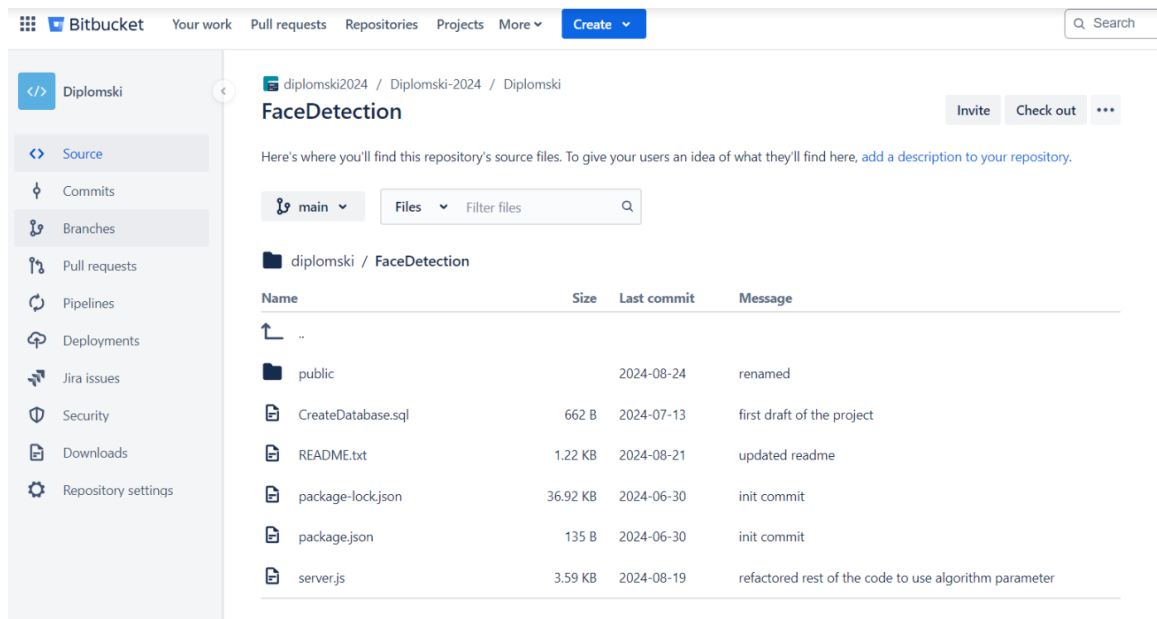
U ovom radu koristi se verzija SQL Server 2020 Standard Edition, a na slici u nastavku prikazana je struktura baze podataka u koju se pohranjuju značajke obrađenih fotografija. Instanca SQL Servera naziva se Katarina\Diplomski2024, a sadrži samo jednu bazu FaceDetection, koja se i koristi za ovaj projekt. Baza je sastavljena od sljedećih tablica: FaceNet_Classes u koju se pohranjuju klase slika koje se šalju na server te FaceNet_Features u koju se ubacuju značajke slika za onu sliku čiji se podatci podudaraju. Na slici se mogu vidjeti podaci koji se nalaze u tablicama nakon što se učitala jedna baza slika u web aplikaciju. Primjerice u bazi postoje dvije različite slike Arona Sorkina te se za svaku sliku ubacila klasa (s obzirom na to da je u našem slučaju klasa ime i prezime, za obje slike je klasa ista). U tablici sa značajkama prikazane su sve značajke za pojedinu sliku kojoj je klasa 'Aron_Sorkin', a pretvorene su u string vrijednost kako bi se mogle pohraniti u SQL Server bazi u obliku podataka NVARCHAR(MAX) koji može pohraniti do jedne milijarde znakova, specijalnih znakova i brojeva zajedno u kombinaciji. Ista stvar napravljena je i s ResNet_Classes i ResNet_Features tablicama, samo što se koristi drugi algoritam za detekciju/identifikaciju slika. Iz tablica ResNet_Features i FaceNet_Features se kasnije čitaju pohranjene značajke za svaku klasu kako bi se identificirala osoba na slici.



Slika 9-15 Struktura baze podataka (Vlastita izrada 12.07.2024.)

9.3.3. BitBucket

Ovaj Atlassian alat služio je za pohranu projekta i izmjena koje su se na istom događale. Repozitorij Diplomski-2024 strukturiran je sa sljedećim mapama u kojima je spreman odgovarajući kod odnosno odgovarajuće skripte. FaceDetection je glavna mapa u kojoj se sve nalazi, a podijeljena je na public mapu u kojoj se u mapu extracted spremaju učitane slike. U tu mapu spremaju se i slike nakon što se detektira lice i dobiju značajke tih slika. Nakon toga se te slike koriste na web aplikaciji kako bi se korisniku prikazala detektirana lica. U CSS mapi nalaze se sve CSS skripte za html stranice. BitBucket je poslužio kao dobar alat za spremanje izmjena tijekom ovog projekta te bih ga svakako, zbog svoje jednostavnosti i lake integracije u Visual Studio Code, ponovno koristila.

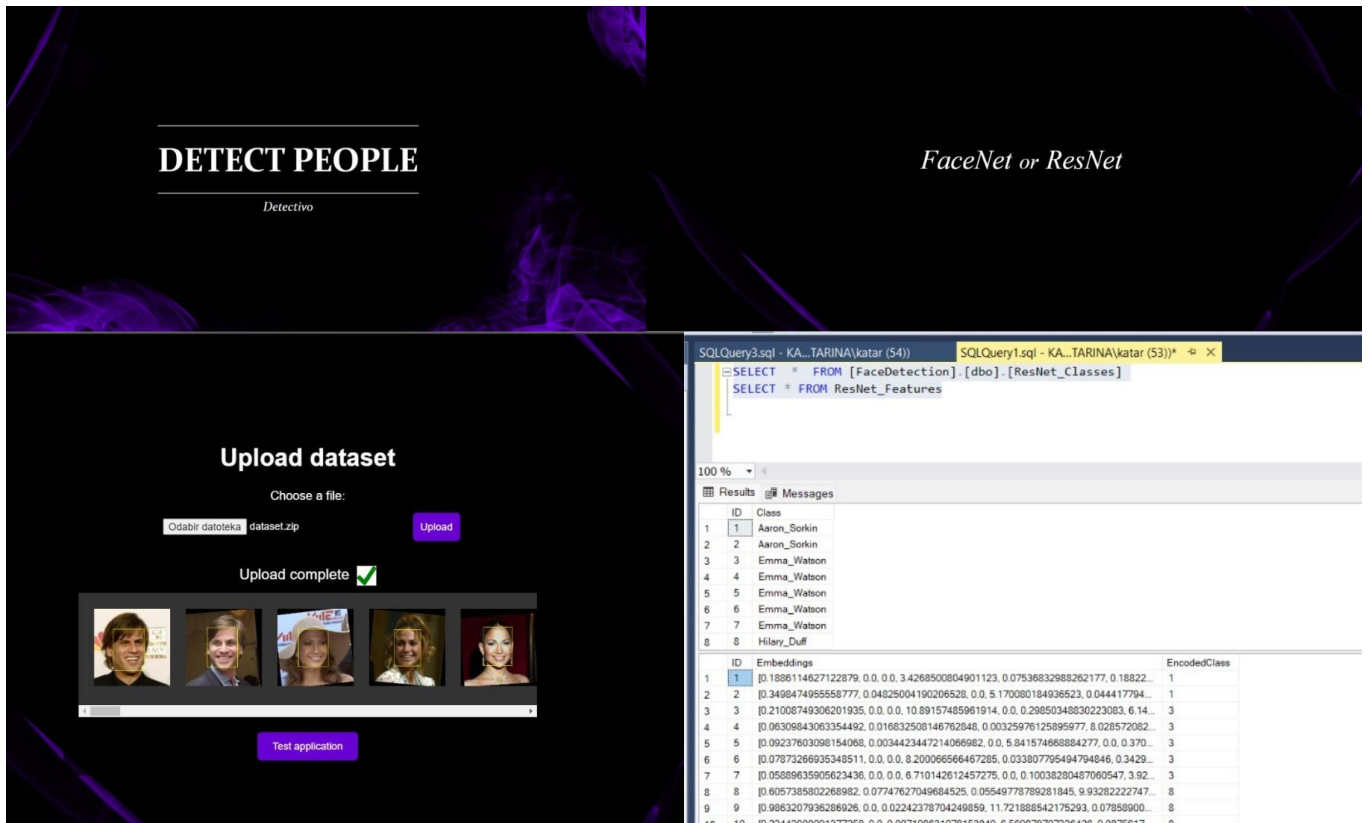


Slika 9-16 Repozitorij projekta (Vlastita izrada 24.08.2024.)

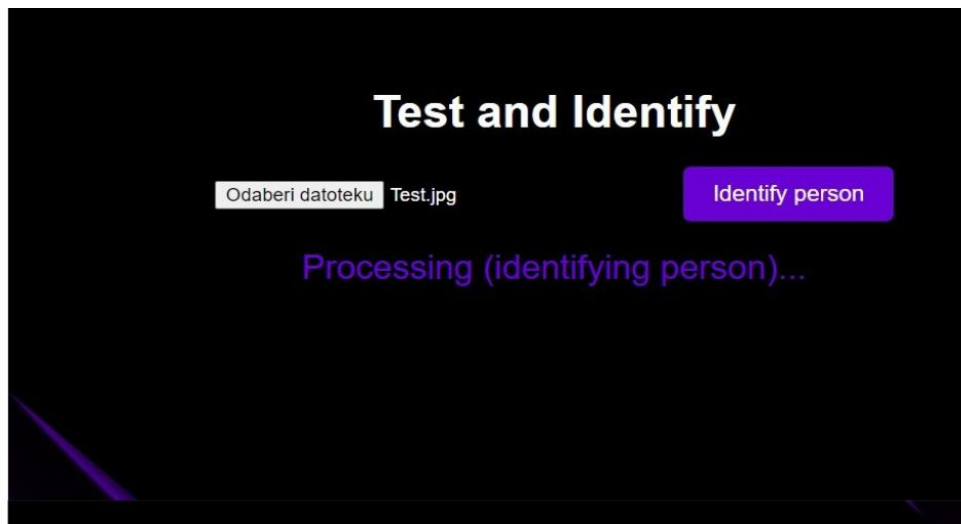
9.4. Hodogram rada aplikacije i opis skripti

Ova se web aplikacija sastoji od nekoliko stranica, a kroz sljedećih šest slika prikazani su koraci u radu aplikacije i kako ju koristiti. Također, [ovdje](#) se nalazi kratki video uradak o radu aplikacije.

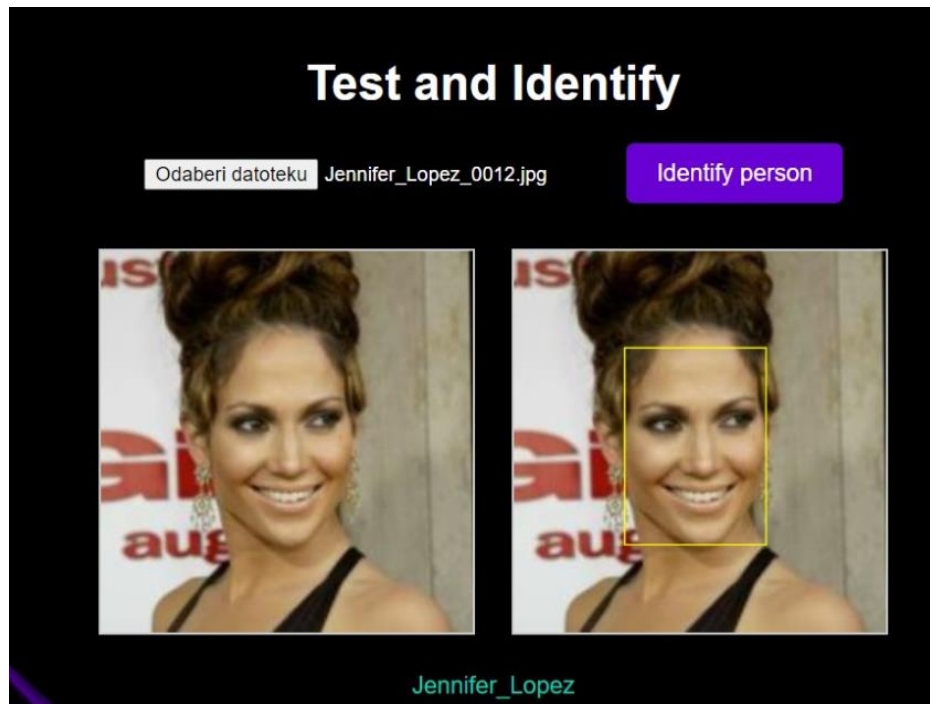
- 1) Početna stranica aplikacije sadrži ime aplikacije *Detectivo*, a odabir na naslov Detect People dolazi se do sljedeće stranice gdje je moguće odabrati koji će se algoritam koristiti, FaceNet ili ResNet (oba koriste MTCNN model za detekciju lica).
- 2) Nakon odabranog algoritma, na sljedećoj stranici potrebno je u .zip formatu predati nekoliko mapa sa slikama (koristi se postojeća baza LFW).
- 3) Algoritam provodi detekciju i ekstrakciju značajki i njihovu pohranu u odgovarajuće tablice u bazi. Ako je sve prošlo uredno, pokazat će se gumb da je detekcija prošla uspješno i prikazat će se detektirana lica na slikama. Također u bazu će se upisati podaci. Nakon toga pojavljuje se gumb "Test Application" koji vodi na sljedeću stranicu gdje je moguće detektirana lica i identificirati.
- 4) Na ovoj stranici potrebno je učitati jednu sliku gdje će se pomoću odgovarajućih klasifikatora i algoritama određeno lice identificirati. Značajke za treniranje modela povlače se iz baze te se on koristi za identifikaciju osobe. Na kraju procesa ispisuje se ime klase koja je u ovom slučaju i samo ime osobe koja je identificirana.



Slika 9-17 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)



Slika 9-18 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)



Slika 9-19 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)

9.4.1. Opis skripti

Kroz ovaj se rad provlači nekoliko programskih i jedan proceduralni jezik te je iz tog razloga teško odvojiti i objasniti skripte prema jeziku koji se koristi s obzirom na to da su kodovi međusobno isprepleteni. Skripte se logički dijele na dva dijela, a to je skripta koja definira server i skripte koje služe za obradu podataka i identificiranje osoba. Također svaka html stranica ima svoju zasebnu skriptu s CSS kodom, također samo zbog preglednosti i lakšeg snalaženja te neće biti objašnjene sve s obzirom na to da se ponavljaju.

Server.js je JavaScript skripta koja je temeljna skripta u ovom projektu. Uz pomoć Node.js servera postavlja web server (Express) na kojem se odvija sav promet i rad vezan uz aplikaciju. Tu su definirane i metode za prijenos i dohvat podataka (GET i POST). GET se koristi za prikaz index.html i ostalih stranica te za vraćanje prepoznatih lica koja se prikazu na stranici (pohranjuju se u mapi public\extraced\detectedFaces). Zatim sam koristila i POST metodu uz pomoć koje se primaju baze i pokreće se odgovarajuća python skripta. POST metoda se na sličan način koristi i za identifikaciju i učitavanje slika na stranici gdje će se identificirati osoba. Isto se pokreću odgovarajuće python skripta (classifier.py).

HandleDataset.py je skripta koja se sastoji od nekoliko bitnijih funkcija i modula koji su mi koristili. Prva koja mi je koristila je reset_folder tako da mi se sa svakim novim učitavanjem

baze ne prebrišu podaci i nastanu problemi te je jednostavnije bilo da se odmah obrišu stare postojeće slike u mapama. Sljedeća je `detect_faces` funkcija koju koristim za detektiranje lica, koja mi vraća parametre X (lica) i Y (klase). Klase su imena mapa u kojima se nalaze slike za pojedinu osobu. Bitna su i `save_to_sq_server_classes` i `save_to_sq_server_features` funkcije koja sprema pročitane klase i izračunate značajke u bazu u tablice `FaceNet_Classes\ResNet_Classes` i `ResNet_Features\FaceNet_Features`. Svakim pozivom skripte postojeći podaci u tablicama se izbrišu, a od SQL komandi tj. funkcija korišten je `cursor` za dohvat i prolazak kroz dohvaćene podatke. Iako `cursor` ima više svojih mana nego prednosti, u ovom konkretnom slučaju meni je poslužio sasvim dovoljno i bolje od `foreach` petlje da prolazi kroz podatke i odmah ih sprema u odgovarajuće tablice. Od modula bitno mi je spomenuti `pathlib` pomoću kojeg funkciji za detekciju lica definiram putanju gdje su lica spremljena, zatim `numpy` koji služi za spremanje značajki i pretvorbu značajki u string vrijednosti kako bi se mogle pohraniti u bazu. `Face_detector` modul koristim za detekciju lica i učitavanje lica sa značajkama, zatim `feature_extractor` za dobivanje značajki i na kraju modul `pyodb` kako bi mogla koristiti SQL upite i povezati se s bazom.

Feature_extractor.py je skripta u kojoj koristim Keras odnosno Tensorflow za dobivanje značajki sa slika, ovisno da li se koristi ResNet50 ili FaceNet. Keras i Tensorflow su alati koji se koriste u dubokom učenju tj. kod treniranja modela, a pokazali su se dobrim u radu s neuronskim mrežama.

Face_detector.py je skripta koja služi za detektiranje lica i spremanje takvih slika s detektiranim licem (okvir u području lica). Za učitavanje slika sam koristila OpenCV, a za detekciju lica MTCNN model.

Face_classifier.py koristim za identifikaciju lica na slikama. Nisu korišteni isti klasifikatori za prepoznavanje osoba zbog znatije hoće li biti kakve veće razlike u radu između algoritama. Za FaceNet sam koristila SVC klasifikator metodom potpornih vektora (engl. *support vector classifier-SVC*), a za ResNet RFC tj. klasifikator metodom slučajne šume (engl. *random forest classifier, RFC*). Klasifikatori su također algoritmi koji služe za klasifikaciju slika, a isto tako sam ih koristila i za predikciju klasa novih slika. SVC je najpopularniji, a RFC nije toliko često korišten, odnosno primjena mu je veća u medicini, klasifikaciji dokumenata i slično, ali je u mom slučaju pokazao da je jako dobar i kod slika te su svi rezultati koji se koriste za identifikaciju na temelju ResNet-a i RFC-a bili također točni.

Index.html je skripta s kodom početne stranice aplikacije gdje se klikom na naziv aplikacije dolazi do stranice za odabir algoritma, a čiji se kod nalazi u skripti **upload.html**. Skripta **uploadDataset.html** sadrži klasični html kod s tipkama, oznakama i ostalim elementima koji su

potrebni za učitavanje slika. **IdentifyPerson.html** napravljena je na sličan način kao i prethodne dvije. Svaka stranica sadrži i svoju CSS skriptu gdje se nalazi dizajn tih stranica.

9.5. ResNet i FaceNet usporedba

U projektu sam implementirala dva pristupa za prepoznavanje lica: jedan koristeći FaceNet, a drugi koristeći ResNet50. FaceNet generira 128-dimenzionalne značajke za svaku sliku lica, dok ResNet50 generira 2048-dimenzionalne značajke. Jednostavno rečeno, ResNet zahtijeva puno više prostora za pohranu tih značajki. S obzirom na to da sam u bazi za kolonu o značajkama koristila tip podatka NVARCHAR(MAX) koji može pohraniti do milijardu znakova, u samoj pohrani u bazu nije bilo razlike. Problem je nastajao već u samom nizu kojeg je u bazu trebalo spremati. U FaceNetu sam koristila np.array funkciju za pretvorbu niza značajki koje je generirao python u string te se bez problema spremalo u bazu. Isto sam pokušala i s ResNetom, no ondje je skripta izbacivala error zato što je kod te pretvorbe velik broj znakova bio izostavljen i skraćen te se nisu pohranila sve značajke. Kako bi se taj problem riješio, dodala sam novu funkciju koja je prvo niz pretvorila u json string te ga kao takvog pohranila u bazu bez gubitka podataka. Prema ovome zaključujem da je FaceNet možda bolji izbor odnosno sigurniji izbor za rad s bazom jer nema straha od izmjene ili gubitka pojedinih podataka. Također, s obzirom na to da ResNet izvlači više značajki za sliku, bilo bi sasvim logično da je i cijeli proces sporiji, no u ovom slučaju to nije bilo tako. Za detekciju lica i pohranu značajki u bazu, FaceNet-u je trebalo 155 sekundi (otprilike 2,5 minute), a ResNet-u 126 sekundi (nešto malo više od 2 minute). Isto tako, za dohvat značajki iz baze i identifikaciju lica, ResNet se opet pokazao kao brža opcija. Za cijeli taj proces nad istom slikom, ResNet je identificirao osobu za 19,5 sekundi, a FaceNet za 24,3 sekunde. Jedini očiti razlog u ovolikoj razlici bio bi korištenje različitih klasifikatora kod identifikacije. Opet bi prema svim čimbenicima FaceNet trebao biti brži, no ipak nije. Razlog tomu je što sam za FaceNet koristila SVC klasifikator, a za ResNet RFC klasifikator. SVC je nešto kompleksnije arhitekture, gdje se kod identificiranja koriste vektori i kod samog treniranja to zahtijeva nešto više vremena zbog matematičkog računanja. RFC koristi takozvana "stabla" koja sadrže značajke slika i može biti nešto brži proces zbog mogućeg paralelizma u izgradnji tih stabala.

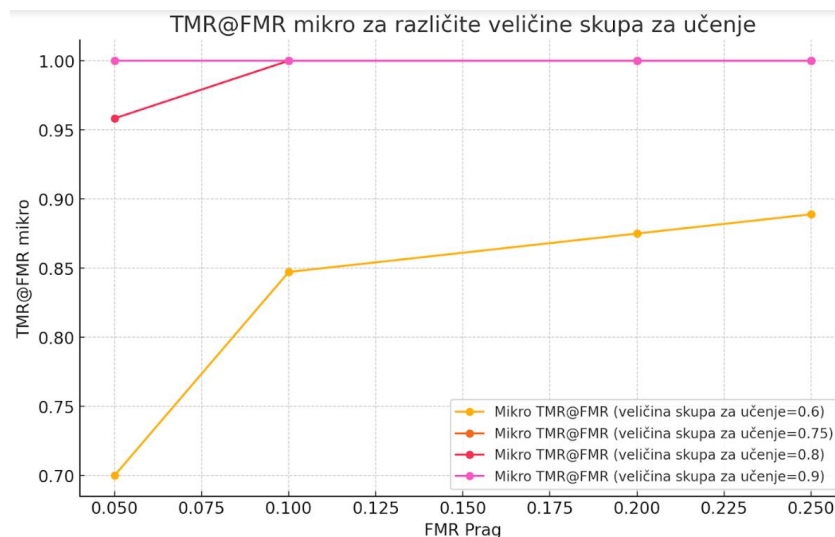
9.6. Testiranje mjera učinkovitosti

Mjere učinkovitosti u radu modela već su detaljno opisane u teorijskom dijelu rada, no testirala sam ih nad svojim modelom i nad bazom koji koristim. Riječ je o bazi s 10 različitih osoba od kojih svaka ima u prosjeku od 2 do 8 slika. Preciznije, cijela baza sastoji se od 72 slike.

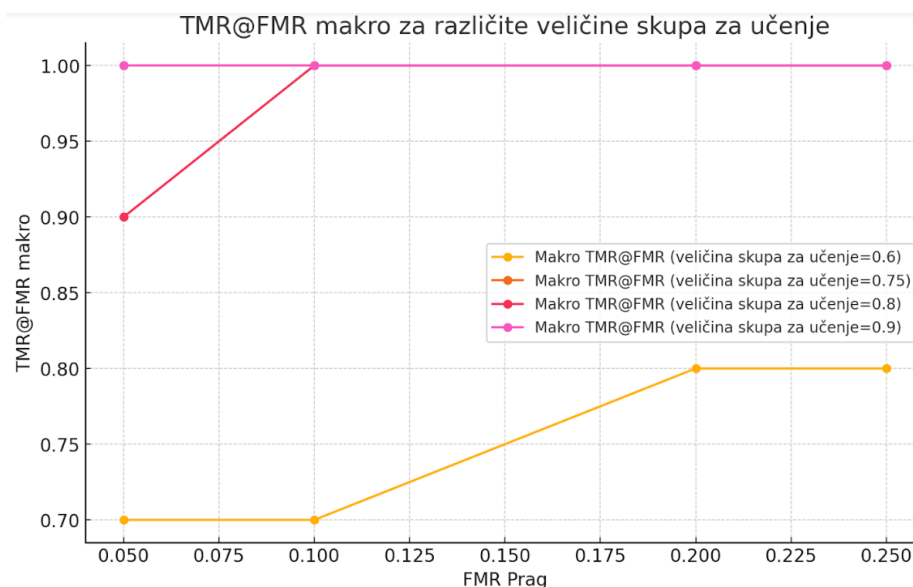
Mjere koje sam testirala su Rank-N te TMR@FMR za ResNet model odnosno algoritam, a rezultati se mogu izračunati pomoću skripte "TestMetrics.py". Kroz mjerenja mijenjani su parametri veličina skupa za treniranje i FMR prag pri čemu na primjer, veličina skupa za treniranje vrijednosti 0.6 predstavlja omjer skupa za treniranje i testiranje kao 0.6:0.4.

9.6.1. FaceNet

Prva metoda testirana jest TMR@FMR odnosno mikro i makro TMR@FMR. Mikro prikuplja podatke iz svih klasa i uzoraka dajući univerzalne rezultate. Makro izračunava TMR za svaku klasu pojedinačno. Krajnji rezultat je prosječna vrijednost svih klasa. Dok su mikro mjerenja manje pouzdana u skupovima podataka čije klase nisu uravnotežene (dobar omjer slika), makro mjerenja su manje osjetljiva na neuravnotežene skupove podataka. U nastavku su prikazani grafovi za mikro i makro TMR@FMR s omjerom skupa za učenje i testiranje od 0.6, 0.75, 0.8, 0.9.



Slika 9-20 FaceNet - mikro TMR@FMR mjerenje (Vlastita izrada 29.8.2024.)

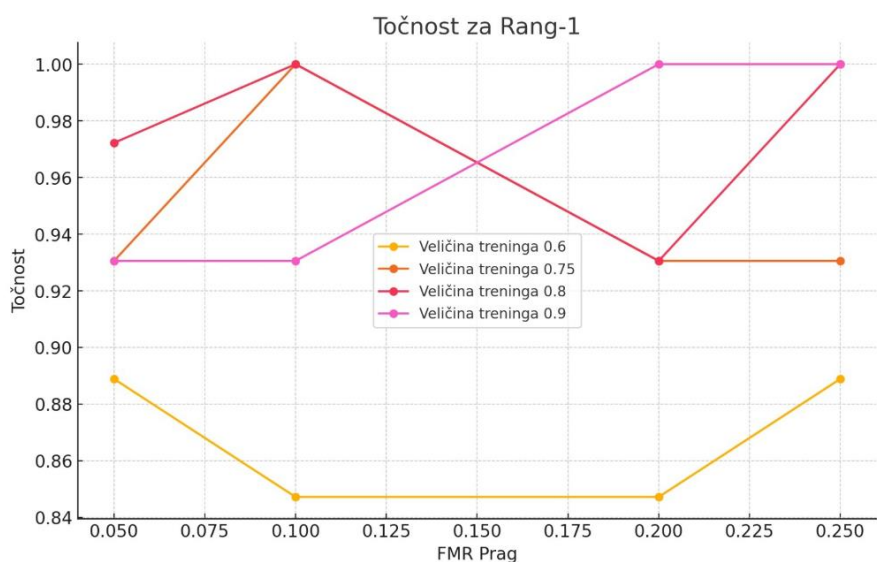


Slika 9-21 FaceNet - makro TMR@FMR mjerenje (Vlastita izrada 29.8.2024)

Iz ovih grafova može se zaključiti da je pozitivni odnosno trend povećavanja TMR-a što zapravo znači da ovaj model ima bolje performanse kada ima veći skup podataka nad kojim može trenirati.

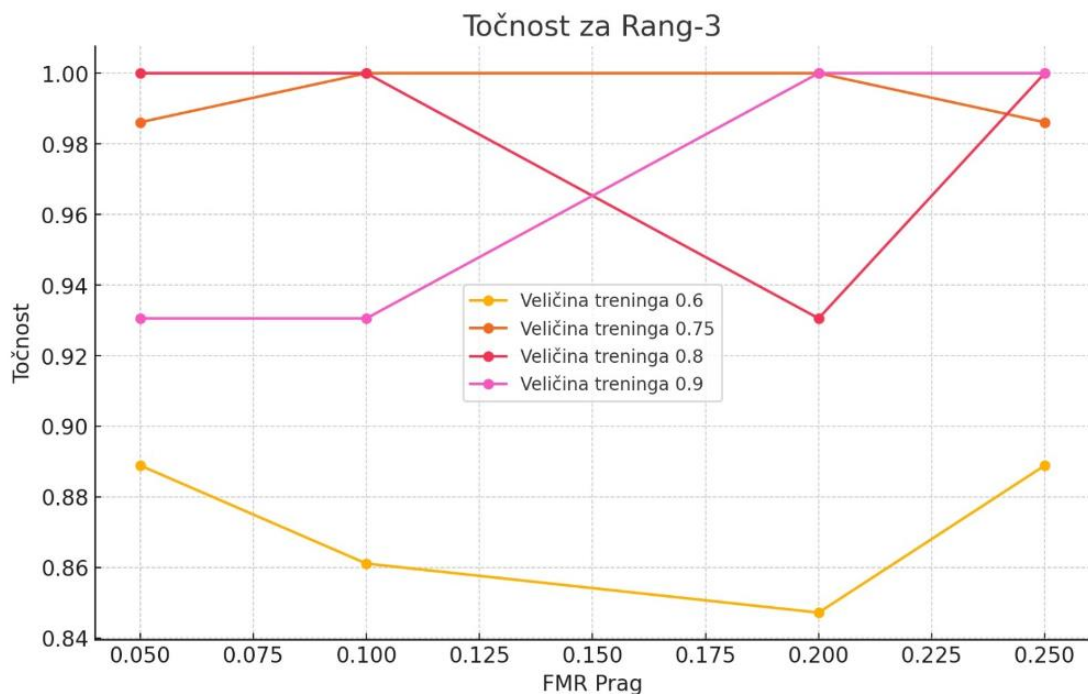
Rank-N je druga testna mjera koja je testirana, a testirana je za $N = 1, 3$ i 5 , dok je omjer skupa za učenje i testiranje opet bio $0.6, 0.75, 0.8$ i 0.9 . Na slikama ispod teksta nalaze se grafovi koji prikazuju omjer točnosti prema FMR pragu i to za rang $1, 3$ ili 5 . Generalni zaključak prema ovim grafovima je da baš kao i u prethodnom slučaju, s povećanjem testnog skupa podataka, i performanse modela postaju bolje.

Za Rang-1 mjerenje također je vidljivo s porastom slika u skupu podataka, rastu i mogućnosti za točnijim predviđanjima. Na primjer, na FMR pragu od $0,05$, točnost se povećava s oko $0,89$ (pri veličini skupa od $0,6$) na približno $0,93$ (pri veličini skupa od $0,9$). Na višim FMR pragovima ($0,20, 0,25$) utjecaj veličine skupa još je izraženiji, s točnošću koja doseže $1,0$ pri većim veličinama skupa za obuku, posebno na FMR pragu od $0,25$.

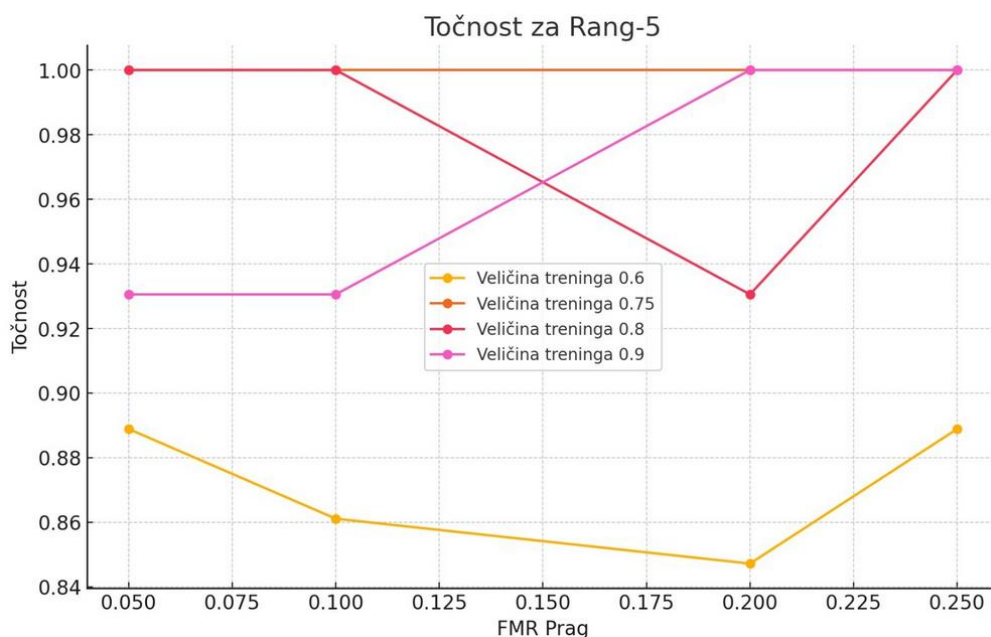


Slika 9-22 FaceNet - Rang-1 mjerenje (Vlastita izrada 20.8.2024)

Točnost za Rang-3 i Rang-5 logički prati rezultate Rang-1 mjerenja, s povećanjem skupa povećava se i točnost modela. Na primjer, pri veličini skupa od 0,6, točnost za Rang-3 poboljšava se FMR 0,05 na FMR 0,25. Pri FMR pragovima od 0,2 i 0,25, točnosti za Rang-3 i Rang-5 dosežu 1,0 što pokazuje da je model vrlo učinkovit u tim uvjetima.



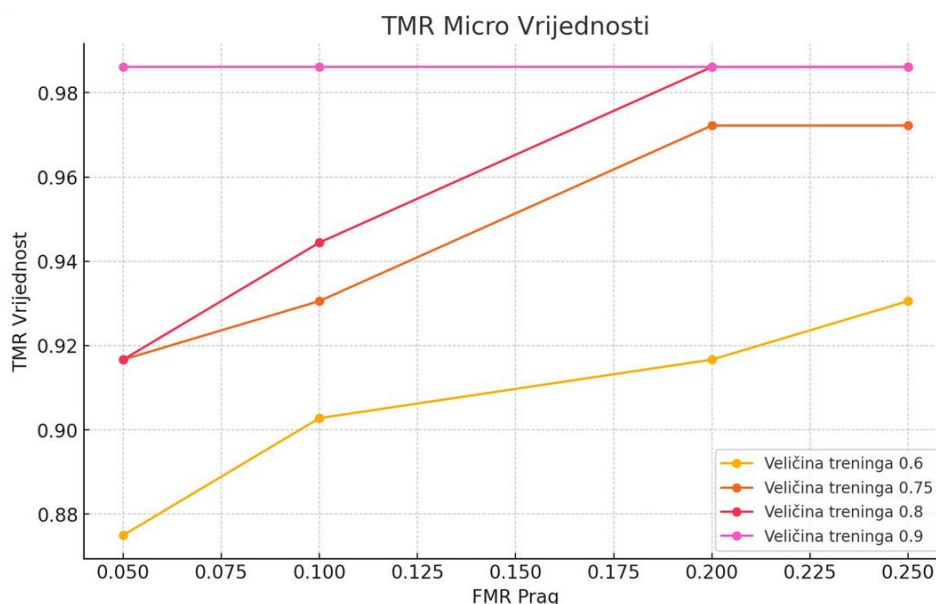
Slika 9-23 FaceNet - Rang-3 mjerenje (Vlastita izrada 20.8.2024)



Slika 9-24 FaceNet - Rang-5 mjerenje (Vlastita izrada 20.8.2024)

9.6.2. ResNet

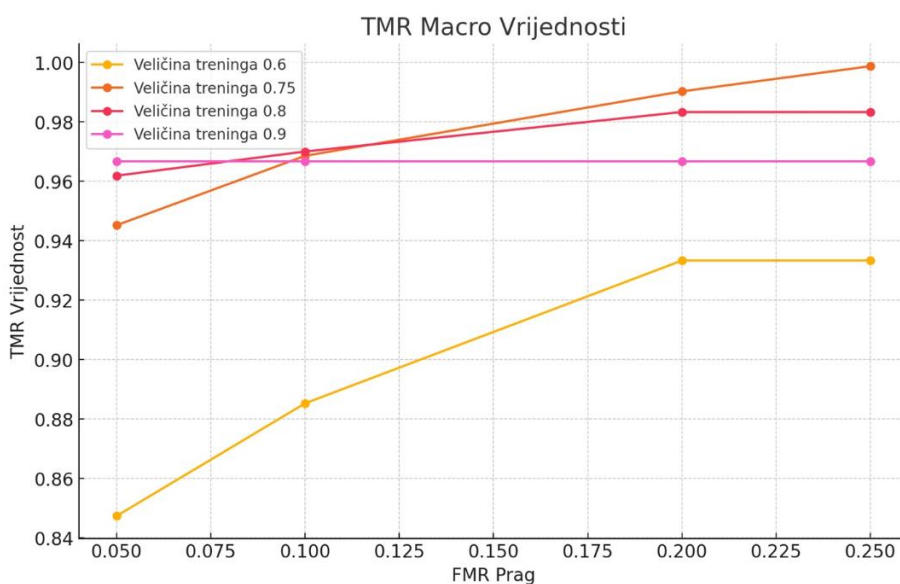
Za isti postupak koji je proveden za FaceNet, napravljena su i mjerenja za ResNet. Mikro i makro TMR@FMR ne odstupaju puno od generalnih rezultata dobivenih korištenjem FaceNet-a. Gledajući širu sliku, zaključak je gotovo isti, a to je da s povećanjem skupa podataka raste i broj točnih predviđanja odnosno raste vrijednost TMR mikro mjerenja. Pragovi korišteni u ovom primjeru su isti: 0.6, 0.75, 0.8 i 0.9. Kod nižih FMR pragova (npr. 0,05), postoji blago manja razlika u TMR mikro vrijednostima između različitih veličina treninga, dok se za veće pragove (npr. 0,25) te razlike povećavaju. Ovo može ukazivati na to da model bolje iskorištava dodatne podatke kako se povećava prag FMR. FMR mikro postiže maksimalnu vrijednost od 0,9861 kada je FMR prag 0,25, bez obzira na veličinu treninga, što pokazuje stabilnost performansi na najvišim pragovima.



Slika 9-25 ResNet - mikro TMR@FMR mjerenje (Vlastita izrada 20.8.2024.)

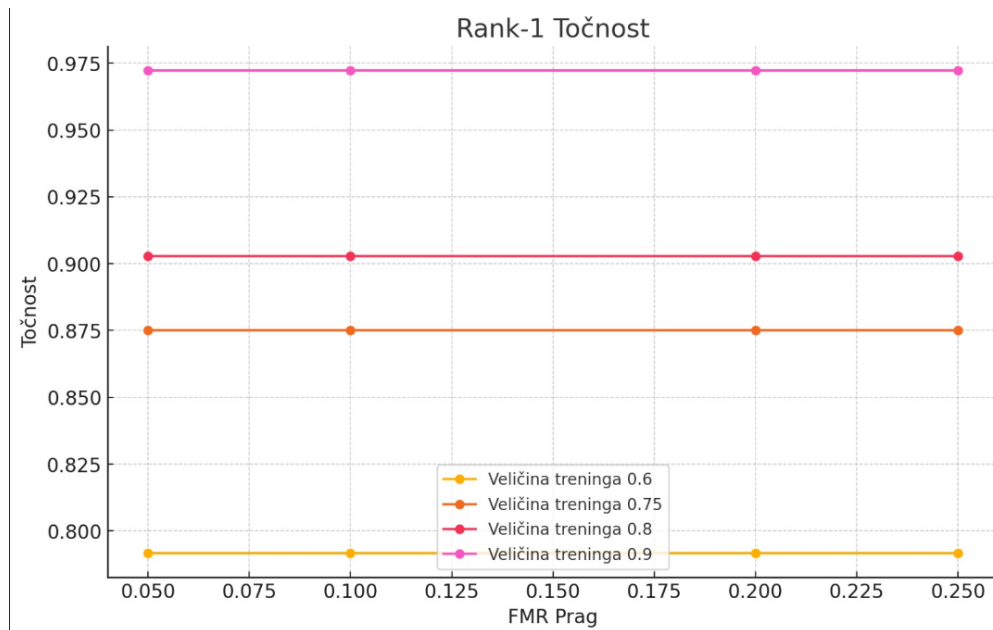
Slično kao kod TMR mikro, TMR makro vrijednosti također pokazuju trend rasta s povećanjem veličine treninga. Međutim, TMR makro ima nešto više varijacija između različitih FMR pragova u usporedbi s TMR Micro. TMR makro vrijednosti su također najveće pri višim FMR pragovima (npr. 0,25), osobito za veće veličine treninga. Kod većih veličina treninga, model pokazuje veći skok u performansama između FMR pragova 0,2 i 0,25.

Najveće TMR makro vrijednosti također su dosegnute na FMR pragu 0,25, pri čemu je maksimalna vrijednost 0,9987 za veličinu treninga 0,75. Veći FMR pragovi omogućuju modelu da postigne bolje TMR rezultate, što je potvrđuje činjenicu da veći prag dopušta veći postotak grešaka u prepoznavanju, ali uz bolju ukupnu točnost.



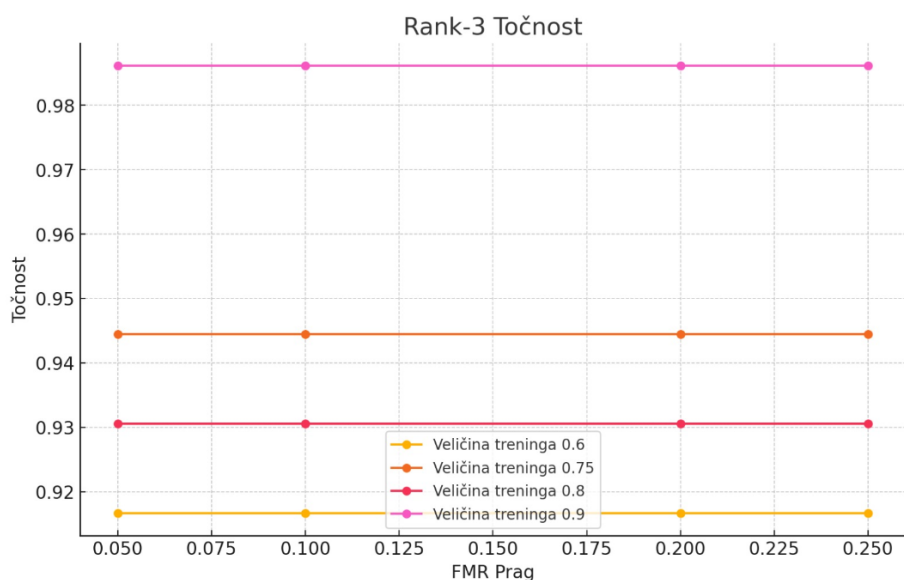
Slika 9-26 ResNet - makro TMR@FMR mjerenje (Vlastita izrada 20.8.2024.)

Rank-N mjerenja provedena su isto kao i za FaceNet za vrijednosti 1, 3 i 5. Rank-1 točnost raste s povećanjem veličine treninga, što je najizraženije kod FMR praga 0,05 i 0,1. Točnost je stabilna i visoka preko različitih FMR pragova, s maksimalnim vrijednostima dosegnutim pri FMR pragu 0,25.



Slika 9-27 ResNet - Rang-1 mjerenje (Vlastita izrada 20.8.2024)

Slično kao kod Rank-1, Rank-3 točnost se poboljšava s povećanjem veličine treninga, iako su promjene manje izražene nego kod Rank-1. Rank-3 točnost pokazuje stabilnost kroz različite FMR pragove, s visokim vrijednostima dosegnutim pri većim veličinama treninga. Rank-5 točnost je gotovo jednaka kao i kod Rank-3, s vrlo visokim vrijednostima pri većim veličinama treninga. Najveće vrijednosti Rank-5 točnosti se također postižu na FMR pragu 0,25, s minimalnim razlikama među veličinama treninga.



Slika 9-28 ResNet - Rang-3 mjerenje (Vlastita izrada 20.8.2024)



Slika 9-29 ResNet - Rang-5 mjerenje (Vlastita izrada 20.8.2024)

Sve tri rang točnosti pokazuju pozitivan utjecaj povećanja veličine treninga, što sugerira da veći set podataka omogućava modelu bolje prepoznavanje točnih rezultata. Sve vrijednosti su stabilne i visoke pri različitim FMR pragovima, pri čemu se najveće vrijednosti dostižu na pragu 0,25. To znači da se model dobro prilagođava različitim pragovima pogrešaka.

10. Zaključak

Detekcija lica i identifikacija osoba na temelju iste, duboko se već uvukla u naše svakodnevne radnje, direktno ili neposredno bez da smo i sami svjesni. Iako se možda čini kao novija i aktualna tema, biometrija je usađena u naše živote već dugi niz godina, a počeci sežu u davne prehistorijske dane. Točna i brza detekcija lica je ono čemu svi teže, no ima svojih mana i izazova, od krivog poravnanja slike, loše kvalitete, nedovoljno dobro treniranih modela i sličnih prepreka. Kako bi rezultati uvijek bili što optimalniji, razvijene su mnoge metode koje se iz dana u dan unaprjeđuju, testiraju i popravljaju. Duboko učenje kao podskup strojnog učenja, omogućava tehnološki razvoj u svim područjima pa tako i u području računalnog vida. Zahvaljujući dubokom učenju razvili su se mnogi modeli za detekciju lica unazad 15-ak godina. Svaka nova verzija YOLO modela sa sobom je donijela noviju, kompleksniju i unaprjeđeniju arhitekturu, na isti se način razvijao i model MobileNet kroz nekoliko verzija, zatim ResNet, VGG, HOG i još mnogi drugi. Bitno je spomenuti da su veliku ulogu u napretku ove industrije imala i neuronske mreže (CNN) odnosno modeli koji se temelje na konvolucijskim neuronskim mrežama poput MTCNN modela koji je korišten i u praktičnom dijelu rada. Prilikom pravilne identifikacije osoba, ti modeli nisu dovoljni te se koriste u kombinaciji s algoritmima za detekciju/identifikaciju. U radu je opisano nekoliko njih poput DeepFace, DeepID i FaceNet algoritma koji je korišten i u identifikacija osoba u web aplikaciji. Kod treniranja svih spomenutih modela, važnu ulogu uvijek imaju i funkcije gubitka koje predstavljaju razliku između predviđenih i stvarnih rezultata modela, a zahvaljujući novijim znanstvenim člancima, opisane su AdaFace koja se zapravo sastoji još od druge dvije funkcije, ArcFace i MagFace gdje svaka sljedeća koristi neke komponente prethodne s unaprijeđenom arhitekturom i što boljim rezultatima. Za potpuno provjeru rada modela i ocjenu njegovih performansi koriste se i mjere učinkovitosti koje prikazuju razne omjere između točnih i netočnih rezultata. Tijekom izrade vlastite web aplikacije za detekciju lica, Rank-N i TMR@FMR su testirane nad podacima koji su korišteni u aplikaciji. Kako bi rad bio temeljit i potkrijepljen praktičnim prikazima svega opisanog, razvijena je web aplikacija *Detectivo* za detekciju i identifikaciju osoba koristeći MTCNN model za detekciju lica te ResNet i FaceNet modele/algoritme za identifikaciju osoba zajedno s različitim klasifikatorima. Dobiveni rezultati su interpretirani i uspoređeni te detaljno opisani u radu.

Detekcija lica postala je komercijalno dostupna industrija, koja ako se koristi u ispravne svrhe može donijeti veliki napredak i sigurnost društvu. Na nama je da iskoristimo tehnologiju na najbolji mogući način te tako svakodnevno razvijamo poboljšanja koja će nam pomoći u svim područjima života gdje se detekcija i identifikacija objekata na slikama koristi.

11. Popis slika

Slika 2-1 Način rada Viola-Jones algoritma za detekciju lica (Preuzeto 5.2.2024. s https://medium.com/@Andrew_D./computer-vision-viola-jones-object-detection-d2a609527b7c)	3
Slika 2-2 Dijagram toka detekcije lica (Vlastita izrada, 20.05.2024.).....	7
Slika 2-3 Primjena algoritama poravnjanja lica (Preuzeto 23.06.2024. s https://link.springer.com/article/10.1007/s11554-021-01107-w/figures/2)	8
Slika 3-1 Podjela podatkovne znanosti (Preuzeto 1.12.2023. s https://tvst.arvojournals.org/article.aspx?articleid=2762344)	12
Slika 3-2 Arhitektura CNN-a (Preuzeto 25.02.2024. s https://www.upgrad.com/blog/basic-cnn-architecture/)	15
Slika 4-1 YOLO metoda kroz godine (Preuzeto 18.02.2024. s https://www.v7labs.com/blog/yolo-object-detection#how-does-yolo-work-yolo-architecture)	17
Slika 4-2 YOLO mreža rešetki i graničnih okvira (Preuzeto 19.02.2024. s https://towardsdatascience.com/evolution-of-yolo-yolo-version-1-afb8af302bd2)	19
Slika 4-3 YOLOv1 primjer arhitekture (Preuzeto 28.06.2024. . Terven ,Cordova-Esparanza 2023.).....	20
Slika 4-4 YOLOv2 arhitektura (Preuzeto 21.06.2024. Terven ,Cordova-Esparanza 2023.).....	21
Slika 4-5 Primjer korištenja YOLOv9 modela (Preuzeto 23.06.2024. s https://viso.ai/computer-vision/yolov9/).....	25
Slika 4-6 Model dubinski odvojive konvolucije (Preuzeto 25.06.2024. s https://viso.ai/deep-learning/xception-model/)	27
Slika 4-7 Primjer korištenja MobileNet metode u zdravstvu (Preuzeto 25.06.2024. s https://viso.ai/deep-learning/mobilenet-efficient-deep-learning-for-mobile-vision/)	28
Slika 4-8 Slojevi MobileNetV2 (Preuzeto 25.06.2024. s https://www.geeksforgeeks.org/mobilenet-v2-architecture-in-computer-vision/)	29
Slika 4-9 Prikaz radnog slijeda MTCNN-a (Preuzeto 27.06.2024. s https://jasonyanglu.github.io/files/lecture_notes/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0_2020/Project/Face%20Recognition%20Based%20on%20MTCNN%20and%20FaceNet.pdf).....	31
Slika 5-1 Rezidualni blok ResNet arhitekture (Preuzeto 26.06.2024. s https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9).....	33

Slika 5-2 VGG arhitektura (Preuzeto 29.06.2024. s https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f)	35
Slika 6-1 Triplet-loss tehnika (Preuzeto 28.06.2024. s https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02)	36
Slika 6-2 Triplet-loss praktični prikaz (Preuzeto 28.06.2024. s https://www.v7labs.com/blog/triplet-loss#what-is-triplet-loss)	37
Slika 6-3 OpenFace primjer korištenja (Preuzeto 28.06.2024. s https://cmusatyalab.github.io/openface/).....	38
Slika 6-4 DeepFace arhitektura (Preuzeto 29.06.2024. s https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/)	39
Slika 6-5 DeepID arhitektura (Preuzeto 29.06.2024. s https://arxiv.org/pdf/2001.07871)	40
Slika 6-6 DeepID arhitektura (Preuzeto 29.06.2024. s https://arxiv.org/pdf/2001.07871)	41
Slika 6-7 VGGFace2 primjer slika (Preuzeto 1.7.2024. s https://ar5iv.labs.arxiv.org/html/1710.08092).....	42
Slika 7-1 Podjela funkcije gubitka (Preuzeto 10.7.2024. s https://heartbeat.comet.ml/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0).....	44
Slika 7-2 Odnos normi značajki i kvalitete slike (Preuzeto 10.7.2024. s https://openaccess.thecvf.com/content/CVPR2022/papers/Kim_AdaFace_Quality_Adaptive_Margin_for_Face_Recognition_CVPR_2022_paper.pdf).....	45
Slika 7-3 ArcFace implementacija (Preuzeto 10.7.2024. s https://openaccess.thecvf.com/content_CVPR_2019/papers/Deng_ArcFace_Additive_Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.pdf)	47
Slika 7-4 Usporedba ArcFace i MagFace funkcije (Pruzeto 10.7.2024. s https://openaccess.thecvf.com/content/CVPR2021/papers/Meng_MagFace_A_Universal_Representation_for_Face_Recognition_and_Quality_Assessment_CVPR_2021_paper.pdf)	49
Slika 9-1 Spajanje na instancu SQL Servera (Vlastita izrada 10.08.2024.)	55
Slika 9-2 Kreiranje baze (Vlastita izrada 10.8.2024.)	56
Slika 9-3 Imenovanje baze (Vlastita izrada 10.8.2024.).....	57
Slika 9-4 Kreiranje tablica (Vlastita izrada 10.8.2024.)	58
Slika 9-5 Imenovanje tablica (Vlastita izrada 10.8.2024.)	59
Slika 9-6 Provjera kreiranih objekata (Vlastita izrada 10.8.2024.)	59
Slika 9-7 Kreiranje druge tablice (Vlastita izrada 10.8.2024.).....	60
Slika 9-8 Odabir "New Query" opcije (Vlastita izrada 10.8.2024.)	60

Slika 9-9 Pokretanje skripte za kreiranje baze i tablica (Vlastita izrada 10.8.2024.)	61
Slika 9-10 Provjera kreirane baze i tablica (Vlastita izrada 10.8.2024.)	61
Slika 9-11 Kloniranje repozitorija (Vlastita izrada 10.8.2024.)	62
Slika 9-12 Struktura kloniranog repozitorija (Vlastita izrada 11.8.2024.)	62
Slika 9-13 Promjena db.connector.py skripte (Vlastita izrada 11.8.2024.)	63
Slika 9-14 Pokretanje servera (Vlastita izrada 25.08.2024.)	63
Slika 9-15 Struktura baze podataka (Vlastita izrada 12.07.2024.)	65
Slika 9-16 Repozitorij projekta (Vlastita izrada 24.08.2024.)	66
Slika 9-17 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)	67
Slika 9-18 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)	67
Slika 9-19 Način rada aplikacije (Vlastite fotografije, 12.07.2024.)	68
Slika 9-20 FaceNet - mikro TMR@FMR mjerenje (Vlastita izrada 29.8.2024.)	71
Slika 9-21 FaceNet - makro TMR@FMR mjerenje (Vlastita izrada 29.8.2024.)	72
Slika 9-22 FaceNet - Rang-1 mjerenje (Vlastita izrada 20.8.2024.)	73
Slika 9-23 FaceNet - Rang-3 mjerenje (Vlastita izrada 20.8.2024.)	73
Slika 9-24 FaceNet - Rang-5 mjerenje (Vlastita izrada 20.8.2024.)	74
Slika 9-25 ResNet - mikro TMR@FMR mjerenje (Vlastita izrada 20.8.2024.)	75
Slika 9-26 ResNet - makro TMR@FMR mjerenje (Vlastita izrada 20.8.2024.)	75
Slika 9-27 ResNet - Rang-1 mjerenje (Vlastita izrada 20.8.2024.)	76
Slika 9-28 ResNet - Rang-3 mjerenje (Vlastita izrada 20.8.2024.)	77
Slika 9-29 ResNet - Rang-5 mjerenje (Vlastita izrada 20.8.2024.)	77

12. Literatura

[1] A. Gillis, „What is facial recognition? | Definition from TechTarget“, Enterprise AI. Pristupljeno: 02. lipanj 2024. [Na internetu]. Dostupno na: <https://www.techtarget.com/searchenterpriseai/definition/facial-recognition>

[2] Hrvatska Enciklopedija, „biometrija - Hrvatska enciklopedija“. Pristupljeno: 02. lipanj 2024. [Na internetu]. Dostupno na: <https://www.enciklopedija.hr/clanak/biometrija>

[3] I. Adjabi, A. Ouahabi, A. Benzaoui, i A. Taleb-Ahmed, „Electronics | Free Full-Text | Past, Present, and Future of Face Recognition: A Review“. Pristupljeno: 05. veljača 2024. [Na internetu]. Dostupno na: <https://www.mdpi.com/2079-9292/9/8/1188>

[4] I. Qadir, „What is the Viola-Jones algorithm?“, Educative. Pristupljeno: 05. veljača 2024. [Na internetu]. Dostupno na: <https://www.educative.io/answers/what-is-the-viola-jones-algorithm>

[5] S. Mayhew, „History of Biometrics | Biometric Update“. Pristupljeno: 02. lipanj 2024. [Na internetu]. Dostupno na: <https://www.biometricupdate.com/201802/history-of-biometrics-2>

[6] „11 use cases for facial recognition“. Pristupljeno: 30. lipanj 2024. [Na internetu]. Dostupno na: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/where-facial-recognition-used>

[7] C. Álvarez Casado i M. Bordallo López, „Real-time face alignment: evaluation methods, training strategies and implementation optimization“, *J. Real-Time Image Process.*, sv. 18, izd. 6, str. 2239–2267, pros. 2021, doi: 10.1007/s11554-021-01107-w.

[8] „What is Face Detection?“, GeeksforGeeks. Pristupljeno: 02. srpanj 2024. [Na internetu]. Dostupno na: <https://www.geeksforgeeks.org/what-is-face-detection/>

[9] V. Shashkina, „Facial Recognition: Applications, Benefits, and Challenges“, ITREX. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://itrexgroup.com/blog/facial-recognition-benefits-applications-challenges/>

[10] S. Singh i S. V. A. V. Prasad, „Techniques and Challenges of Face Recognition: A Critical Review“, *Procedia Comput. Sci.*, sv. 143, str. 536–543, sij. 2018, doi: 10.1016/j.procs.2018.10.427.

[11] A. Mehta, „The Threat of Facial Recognition Technology“. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://shorturl.at/HSZz0>

[12] Thales Blog Post, „Top facial recognition issues and concerns“. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/facial-recognition-issues>

[13] J. McCarthy, „What is Artificial Intelligence“. Standofrd University, 11. prosinac 2007. Pristupljeno: 01. prosinac 2023. [Na internetu]. Dostupno na: <chrome-extension://efaidnbmninnibpcapjpcglclefindmkaj/https://www-formal.stanford.edu/jmc/whatisai.pdf>

[14] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, i J. P. Campbell, „Introduction to Machine Learning, Neural Networks, and Deep Learning“, *Transl. Vis. Sci. Technol.*, sv. 9, izd. 2, Art. izd. 2, velj. 2020, doi: 10.1167/tvst.9.2.14.

[15] K. Reyes, „What is Deep Learning and How Does It Works [Updated]“, Simplilearn.com. Pristupljeno: 05. veljača 2024. [Na internetu]. Dostupno na: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning>

[16] B. Mayank, „What is Neural Network: Overview, Applications, and Advantages“, Simplilearn.com. Pristupljeno: 05. veljača 2024. [Na internetu]. Dostupno na: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-neural-network>

[17] T. Beysolow Ii, *Introduction to Deep Learning Using R*. Berkeley, CA: Apress, 2017. doi: 10.1007/978-1-4842-2734-3. Pristupljeno: 27. srpanj 2024. Dostupno na : <http://link.springer.com/10.1007/978-1-4842-2734-3>

[18] L. Deng i Y. Liu, *Deep Learning in Natural Language Processing*. Springer, 2018. Pristupljeno: 29. lipanj 2024. Dostupno na : <https://link.springer.com/book/10.1007/978-981-10-5209-5>

[19] A. K. Tyagi, A. Pramod, i H. S. Naicker, „Machine Learning and Deep Learning: Open Issues and Future Research Directions for the Next 10 Years - Computational Analysis and Deep Learning for Medical Care - Wiley Online Library“. Pristupljeno: 05. veljača 2024. [Na internetu]. Dostupno na: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119785750.ch18>

[20] I. Goodfellow, Y. Bengio, i A. Courville, „Deep Learning“. Pristupljeno: 27. veljača 2024. [Na internetu]. Dostupno na: <https://www.deeplearningbook.org/>

[21] Y. LeCun i X. Zhang, „Text Understanding from Scratch“, ar5iv. Pristupljeno: 27. veljača 2024. [Na internetu]. Dostupno na: <https://ar5iv.labs.arxiv.org/html/1502.01710>

[22] K. Zhang, Z. Zhang, Z. Li, i Y. Qiao, „Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks“, *IEEE Signal Process. Lett.*, sv. 23, tra. 2016, doi: 10.1109/LSP.2016.2603342. Pristupljeno 1. srpanj 2024. [Na internetu]. Dostupno na : <http://ieeexplore.ieee.org/document/7553523/>

[23] D. R. Yehoshua, „Backpropagation: Step-By-Step Derivation“, Medium. Pristupljeno: 27. veljača 2024. [Na internetu]. Dostupno na: <https://towardsdatascience.com/backpropagation-step-by-step-derivation-99ac8fbdcc28>

[24] R. Girshick, J. Donahue, T. Darrell, i J. Malik, „Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation“, stu. 2023, [Na internetu]. Dostupno na: https://www.researchgate.net/publication/258374356_Rich_Feature_Hierarchies_for_Accurate_Object_Detection_and_Semantic_Segmentation

[25] L. Wolf, M. Ranzato, M. Yang, i Y. Taigman, „DeepFace: Closing the Gap to Human-Level Performance in Face Verification“. Tel Aviv University, 23. kolovoz 2014. [Na internetu]. Dostupno na: https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf

[26] J. Redmon, S. Divvala, R. Girshick, i A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“, 09. svibanj 2016., *arXiv*: arXiv:1506.02640. doi: 10.48550/arXiv.1506.02640. Dostupno na : <http://arxiv.org/abs/1506.02640>

[27] J. Redmon i A. Farhadi, „YOLO9000: Better, Faster, Stronger“, 25. prosinac 2016., *arXiv*: arXiv:1612.08242. doi: 10.48550/arXiv.1612.08242. Dostupno na: <http://arxiv.org/abs/1612.08242>

[28] J. Krapac i S. Šegvić, „Konvolucijski modeli“. [Na internetu]. Pristupljeno 1. srpanj 2024. Dostupno na: <https://www.zemris.fer.hr/~ssegvic/du/du2convnet.pdf>

[29] J. R. Terven i D. M. Cordova-Esparanza, „A Comprehensive Review of YOLO From YOLOv1 and Beyond.pdf“, Google Docs. Pristupljeno: 05. srpanj 2024. [Na internetu]. Dostupno na: https://drive.google.com/file/d/1tpoReSY8mIewvBffMkGpcS4vN0MWOXXb/view?usp=drive_open&edoph=true&usp=embed_facebook

[30] J. Huber, „Batch normalization in 3 levels of understanding“, Medium. Pristupljeno: 05. srpanj 2024. [Na internetu]. Dostupno na: <https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338>

[31] S. Rallabandi, „Activation functions: ReLU vs. Leaky ReLU“, Medium. Pristupljeno: 05. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/@sreeku.ralla/activation-functions-relu-vs-leaky-relu-b8272dc0b1be>

[32] G. Boesch, „YOLOv9: Advancements in Real-time Object Detection (2024)“, viso.ai. Pristupljeno: 05. srpanj 2024. [Na internetu]. Dostupno na: <https://viso.ai/computer-vision/yolov9/>

[33] N. Klingler, „MobileNet - Efficient Deep Learning for Mobile Vision“, viso.ai. Pristupljeno: 06. srpanj 2024. [Na internetu]. Dostupno na: <https://viso.ai/deep-learning/mobilenet-efficient-deep-learning-for-mobile-vision/>

[34] G. Boesch, „Xception Model: Analyzing Depthwise Separable Convolutions“, viso.ai. Pristupljeno: 06. srpanj 2024. [Na internetu]. Dostupno na: <https://viso.ai/deep-learning/xception-model/>

[35] N. Sharma, „What is MobileNetV2? Features, Architecture, Application and More“, Analytics Vidhya. Pristupljeno: 06. srpanj 2024. [Na internetu]. Dostupno na: <https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2/>

[36] A. Howard et al., „Searching for MobileNetV3“, u *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, lis. 2019, str. 1314–1324. doi: 10.1109/ICCV.2019.00140. Pristupljeno: 1. kolovoz 2024. Dostupno na: <https://shorturl.at/ojoox>

[37] L. Zhao i L. Wang, „A new lightweight network based on MobileNetV3“, *KSII Trans. Internet Inf. Syst. TIIS*, sv. 16, izd. 1, str. 1–15, 2022, doi: 10.3837/tiis.2022.01.001. Pristupljeno 17. srpanj 2024. Dostupno na: <https://shorturl.at/i5VW5>

[38] R. Jin, H. Li, J. Pan, W. Ma, i J. Lin, „Face Recognition Based on MTCNN and FaceNet“. Pristupljeno 9. srpanj 2024. Dostupno na : <https://shorturl.at/urqo7>

[39] I. Azeem, „Understanding ResNet Architecture: A Deep Dive into Residual Neural Network“, Medium. Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9>

[40] S. Rahul, „Can you tell us something about ‘Global Average Pooling’?“, Aaweg Interview. Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/aaweg-i-nterview/computer-vision-global-average-pooling-d2bbdfbd713f>

[41] A. Kouidri, „Mastering ResNet: Deep Learning Breakthrough in Image Recognition“. Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <https://www.ikomia.ai/blog/mastering-resnet-deep-learning-image-recognition>

[42] S. Bangar, „VGG-Net Architecture Explained“, Medium. Pristupljeno: 08. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>

[43] D. Shah, „Triplet Loss: Intro, Implementation, Use Cases“. Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <https://www.v7labs.com/blog/triplet-loss>, <https://www.v7labs.com/blog/triplet-loss>

[44] K. Dhanush, „facenet-pytorch | Pretrained Pytorch face detection (MTCNN) and facial recognition...“, Medium. Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/@danushidk507/facenet-pytorch-pretrained-pytorch-face-detection-mtcnn-and-facial-recognition-b20af8771144>

[45] „OpenFace | MultiComp“, MultiComp | MultiComp Lab’s mission is to build the algorithms and computational foundation to understand the interdependence between human verbal, visual, and vocal behaviors expressed during social communicative interactions.

Pristupljeno: 07. srpanj 2024. [Na internetu]. Dostupno na: <http://multicomp.cs.cmu.edu/resources/openface/>

[46] „OpenFace“. Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na: <https://cmusatyalab.github.io/openface/>

[47] K. Hao, „This is how we lost control of our faces“, MIT Technology Review. Pristupljeno: 08. srpanj 2024. [Na internetu]. Dostupno na: <https://www.technologyreview.com/2021/02/05/1017388/ai-deep-learning-facial-recognition-data-history/>

[48] S. Serengil, „DeepFace: A Popular Open Source Facial Recognition Library“, viso.ai. Pristupljeno: 08. srpanj 2024. [Na internetu]. Dostupno na: <https://viso.ai/computer-vision/deepface/>

[49] J. M. Morgan, „DeepFace, Facial Verification Software Developed at Facebook, Approaches Human Ability: History of Information“. [Na internetu]. Dostupno na: <https://historyofinformation.com/detail.php?id=4081>

[50] CIW Team, „Chinese University of Hong Kong DeepID Face Recognition Tech Beats Facebook Deepface“, China Internet Watch. Pristupljeno: 08. srpanj 2024. [Na internetu]. Dostupno na: <https://www.chinainternetwatch.com/7757/chinese-university-of-hong-kong-deepid-face-recognition-tech-beats-facebook-deepface/>

[51] S. Shahsavarani, M. Analoui, i R. S. Ghiass, „**M2**Deep-ID: A Novel Model for Multi-View Face Identification Using Convolutional Deep Neural Networks“. Pristupljeno 7. kolovoz 2024. Dostupno na : <https://arxiv.org/pdf/2001.07871>

[52] T. Wood, „Softmax Function“, DeepAI. Pristupljeno: 08. srpanj 2024. [Na internetu]. Dostupno na: <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>

[53] C. Qiong, L. Shen, W. Xie, i O. M. Parkhi, „VGGFace2: A dataset for recognising faces across pose and age“. University of Oxford, 2018. Pristupljeno 5. srpanj 2024. Dostupno na: <https://www.robots.ox.ac.uk/~vgg/publications/2018/Cao18/cao18.pdf>

[54] R. Alake, „Loss Functions in Machine Learning Explained“. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://www.datacamp.com/tutorial/loss-function-in-machine-learning>

[55] Blog post Medium, „14 Loss functions you can use for Regression“, Medium. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://medium.com/@mlblogging.k/14-loss-functions-you-can-use-for-regression-b24db8dff987>

[56] Alpha Capitalis, Blog Post, „Koeficijent korelacije“, Alpha Capitalis. Pristupljeno: 10. srpanj 2024. [Na internetu]. Dostupno na: <https://alphacapitalis.com/glossary/koeficijent-korelacije/>

[57] M. Kim, A. K. Jain, i X. Liu, „AdaFace: Quality Adaptive Margin for Face Recognition“, u *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, lip. 2022, str. 18729–18738. doi: 10.1109/CVPR52688.2022.01819. Pristupljeno 2. srpanj 2024. Dostupno na: https://openaccess.thecvf.com/content/CVPR2022/papers/Kim_AdaFace_Quality_Adaptive_Margin_for_Face_Recognition_CVPR_2022_paper.pdf

[58] J. Deng, J. Guo, N. Xue, i S. Zafeiriou, „ArcFace: Additive Angular Margin Loss for Deep Face Recognition“, 2018. Pristupljeno 14. srpanj 2024. Dostupno na: https://openaccess.thecvf.com/content_CVPR_2019/papers/Deng_ArcFace_Additive_Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.pdf

[59] Q. Meng, S. Zhao, Z. Huang, i F. Zhou, „MagFace: A Universal Representation for Face Recognition and Quality Assessment“, u *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, lip. 2021, str. 14220–14229. doi: 10.1109/CVPR46437.2021.01400. Pristupljeno 11. srpanj 2024. Dostupno na: https://openaccess.thecvf.com/content/CVPR2021/papers/Meng_MagFace_A_Universal_Representation_for_Face_Recognition_and_Quality_Assessment_CVPR_2021_paper.pdf

[60] V. De Mel, „Survey of Evaluation Metrics in Facial Recognition Systems“, ResearchGate. Pristupljeno: 09. srpanj 2024. [Na internetu]. Dostupno na: https://www.researchgate.net/publication/372232460_Survey_of_Evaluation_Metrics_in_Facial_Recognition_Systems



IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, KATARINA ANTONIĆ pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autorica diplomskog rada pod naslovom DETEKCIJA I PREPOZNAVANJE ILIKA KORISTEĆI DUBOKO UČENJE I SVA BAZA ZA POHRANU te da unavedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:

KATARINA ANTONIĆ
(vlastoručni potpis)

Katarina Antonić

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.