Teorijska analiza stabilnosti konvolucijskih neuronskih mreža

Matišić, Petar

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Petar Matišić

TEORIJSKA ANALIZA STABILNOSTI KONVOLUCIJSKIH NEURONSKIH MREŽA

DIPLOMSKI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Petar Matišić

Matični broj: 0016145882

Studij: Informacijsko i programsko inženjerstvo

TEORIJSKA ANALIZA STABILNOSTI KONVOLUCIJSKIH NEURONSKIH MREŽA

DIPLOMSKI RAD

Mentorica:

izv. prof. dr. sc. Petra Grd

Varaždin, rujan 2024.

Petar Matišić

Izjava o izvornosti

Izjavljujem da je ovaj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI Radovi

Sažetak

Ovaj diplomski rad fokusira se na teorijsku analizu stabilnosti konvolucijskih neuronskih mreža (CNN) s posebnim naglaskom na njihove reakcije na ulazne promjene. Stabilnost CNN-a od suštinskog je značaja za njihovu pouzdanost u praktičnim primjenama. Rad započinje pregledom osnovnih komponenti CNN-a, uključujući arhitekturu i funkcioniranje konvolucijskih slojeva. Slijedi definiranje stabilnosti i robusnosti CNN-a, uz analizu matematičkih osnova stabilnosti kroz Lipschitzove konstante i srodne metrike. Istražuju se različite vrste ulaznih promjena te njihov utjecaj na performanse mreža. Metodološki pristup kombinira matematičke modele s eksperimentalnim metodama za procjenu stabilnosti i robusnosti. U empirijskom dijelu rada koristi se EfficientNet model za prepoznavanje dobi i spola na skupovima podataka IMDB i Wiki. Modeli se treniraju i evaluiraju kako bi se analizirale njihove performanse, stabilnost i robusnost. Rezultati uključuju detaljnu statističku analizu predikcija i performansi, kao i analizu međukoraka u procesu treniranja. Zaključci rada pružaju sveobuhvatan uvid u teorijske i empirijske analize, identificiraju ključne doprinose te predlažu smjernice za buduća istraživanja u ovom području.

Ključne riječi: konvolucijske neuronske mreže; stabilnost; robusnost; ulazne promjene; matematičke metode; statistička analiza; lice

Sadržaj

BOGU NA ČAST.

1. Uvod

1.1. Opis problema i motivacija

Konvolucijske neuronske mreže (CNN) predstavljaju temelj modernog računalnog vida, omogućujući napredne sposobnosti u obradi i interpretaciji vizualnih podataka. Njihova široka primjena u područjima kao što su autonomna vozila, medicinska dijagnostika i sustavi za prepoznavanje lica naglašava njihovu važnost u suvremenom društvu. Stabilnost CNN-a, odnosno njihova sposobnost da ostanu pouzdane i točne unatoč malim promjenama u ulaznim podacima, ključna je za njihove praktične primjene. U realnim scenarijima, ulazni podaci često su izloženi raznim vrstama perturbacija, poput šuma (buke) ili namjernih adversarijalnih napada, što može značajno utjecati na performanse mreže. Primjerice, mala promjena u slici prometnog znaka može dovesti do pogrešne klasifikacije, što bi moglo imati ozbiljne posljedice u kontekstu autonomne vožnje. Stoga, razumijevanje i analiza stabilnosti CNN-a postaju od suštinske važnosti za unapređenje njihovih sposobnosti i robusnosti. Ovaj rad motiviran je potrebom za dubljim teorijskim razumijevanjem stabilnosti CNN-a i razvijanjem metoda za njihovu procjenu i unapređenje. Analizom postojećih pristupa i razvijanjem novih metodologija, ovaj rad nastoji pridonijeti povećanju pouzdanosti i sigurnosti sustava temeljenih na CNN-u.

1.2. Ciljevi i struktura rada

Ciljevi ovog rada su višestruki. Prvo, teorijski analizirati stabilnost konvolucijskih neuronskih mreža, koristeći matematičke modele i formalne metode. Ovo uključuje proučavanje matematičkih svojstava CNN-a, poput Lipschitzovih konstanti, koje omogućuju procjenu kako male promjene u ulazu utječu na izlazne rezultate mreže. Drugo, istražiti utjecaj različitih vrsta ulaznih perturbacija na performanse CNN-a te identificirati ključne faktore koji doprinose stabilnosti i robusnosti mreža. Ovdje će se razmotriti različite vrste šuma (buke) i adversarijalnih napada te kako oni mogu destabilizirati mrežu. Treće, provesti empirijske analize koristeći EfficientNet arhitekturu CNN-a na zadacima prepoznavanja dobi i spola, kako bi se evaluirale njihove performanse i stabilnost pod različitim uvjetima. Korištenjem skupova podataka kao što su IMDB i Wiki, model će biti treniran i evaluiran kako bi se dobio uvid u njegovu stabilnost i robusnost. Konačno, rad će pružiti preporuke za buduća istraživanja u cilju daljnjeg poboljšanja stabilnosti i otpornosti CNN-a kroz identificiranje ključnih izazova i potencijalnih rješenja.

Ovaj rad strukturiran je na sljedeći način. U prvom poglavlju, Uvod, iznosi se opis problema, motivacija, ciljevi rada i struktura samog rada. Drugo poglavlje, Teorijska pozadina, pokriva osnovne koncepte i arhitekturu konvolucijskih neuronskih mreža, definira stabilnost i robusnost, te analizira utjecaj ulaznih perturbacija na performanse mreža. U ovom poglavlju detaljno se obrađuju metode za procjenu stabilnosti i robusnosti, uključujući matematičke i eksperimentalne pristupe. Treće poglavlje, Analiza skupova podataka, fokusira se na pregled odabranih skupova podataka, njihove pripreme i čišćenje. Skupovi podataka kao što su IMDB i Wiki bit će analizirani kako bi se dobili čisti i relevantni podaci za treniranje modela. Četvrto

poglavlje, Primjena modela na prepoznavanje dobi i spola, opisuje odabir modela, treniranje i evaluaciju performansi, stabilnosti i robusnosti. Arhitektura poput EfficientNet bit će korištena kako bi se procijenila njena učinkovitost u zadacima prepoznavanja. Peto poglavlje, Rezultati i analiza, pruža detaljnu statističku analizu predikcija, performansi i međukoraka u treniranju. Ovdje će se prikazati rezultati dobiveni tokom empirijskih istraživanja te njihova interpretacija u kontekstu teorijskih nalaza. Zaključno poglavlje sažima istraživanje, identificira glavne doprinose rada i daje preporuke za buduća istraživanja. Ovaj dio rada osvrnut će se na postignute rezultate te pružiti smjernice za daljnji razvoj u području stabilnosti konvolucijskih neuronskih mreža.

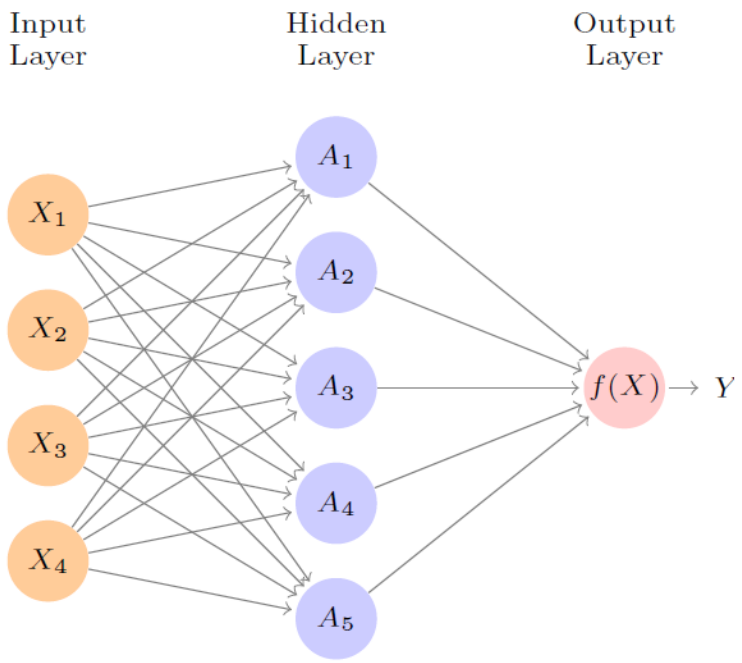
2. Teorijska pozadina

Ovo poglavlje predstavlja teorijsku osnovu za analizu stabilnosti konvolucijskih neuronskih mreža (CNN, *Convolutional Neural Networks*). Prvi dio poglavlja fokusira se na osnovne koncepte i arhitekturu CNN-a, uključujući ključne elemente poput konvolucijskih i *pooling* slojeva te njihov način funkcioniranja. U nastavku se razmatraju stabilnost i robusnost CNN-a, pri čemu se definiraju osnovni pojmovi i iznose matematičke osnove stabilnosti, kao i faktori koji utječu na stabilnost ovih mreža. Zatim se obrađuju različite vrste ulaznih promjena i njihovi učinci na performanse CNN-a. Ovaj teorijski pregled postavlja temelj za razumijevanje izazova i rješenja u području stabilnosti CNN-a, što je ključno za daljnji razvoj primjena u računalnom vidu.

2.1. Osnove konvolucijskih neuronskih mreža (CNN)

2.1.1. Arhitektura

Konvolucijske neuronske mreže (CNN, *Convolutional Neural Networks*) su jedna od najvažnijih arhitektura u domeni dubokog učenja, posebno prilagođene za rad s podacima koji imaju prostornu strukturu, kao što su slike i videozapisi [1, str. 331]. CNN-i su osmišljeni kako bi iskoristili prostorne relacije unutar podataka koristeći lokalne povezanosti, smanjenje dimenzionalnosti i hijerarhijsko učenje značajki [1, str. 332].



Slika 1: Neuronska mreža sa jednim skrivenim slojem (*hidden layer*); preuzeto iz [2, str. 400]

Da bi se razumio koncept CNN, prvo je potrebno razumjeti osnove jednostavne neuronske mreže s jednim skrivenim slojem, kao što je prikazano na slici 1. Jednostavna neuronska mreža sastoji se od ulaznog sloja (*input layer*), jednog ili više skrivenih slojeva (*hidden layers*) i izlaznog sloja (*output layer*). Svaki sloj sastoji se od neurona koji su međusobno povezani težinama (*weights*), koje se prilagođavaju tijekom procesa učenja [2, str. 400, 401]. Matematički model jednostavne neuronske mreže može se izraziti kao:

$$z_j = \sum_{i=1}^{p} x_i w_{ij} + b_j$$

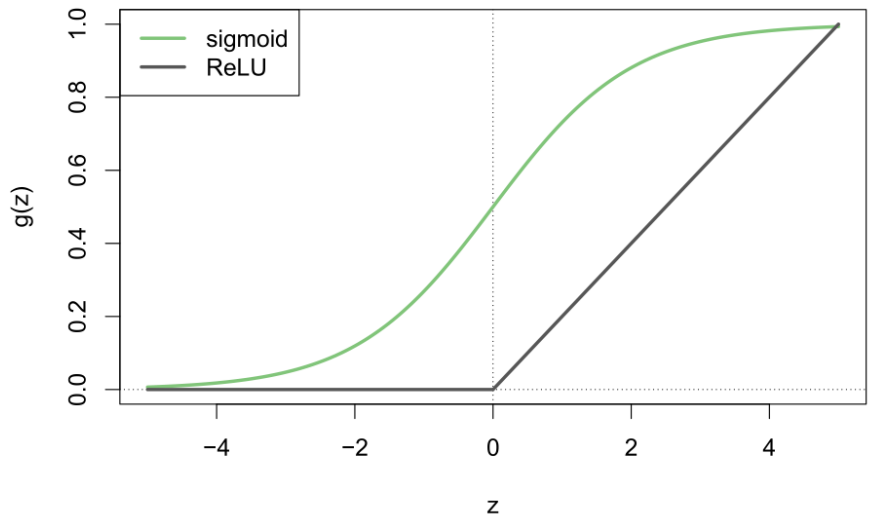
gdje je z_j aktivacija neurona u skrivenom sloju j, x_i su ulazne vrijednosti, w_{ij} su težine povezane s ulazom i i neuronom j, a b_j je pristranost (bias) neurona j [2, str. 401]. Nakon što se izračuna linearna kombinacija ulaznih podataka i težina, aktivacija z_j prolazi kroz aktivacijsku funkciju σ, koja uvodi nelinearnost u mrežu [1, str. 194]:

$$a_j = \sigma(z_j)$$

Aktivacijska funkcija može biti, na primjer, *sigmoid* funkcija, koja je definirana kao:

$$g(z) = \text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid funkcija preslikava bilo koju stvarnu vrijednost u interval između 0 i 1, što omogućava interpretaciju rezultata kao vjerojatnosti [1, str. 194]. U kontekstu klasifikacije, rezultat funkcije na izlazu može se interpretirati kao vjerojatnost pripadnosti jednoj od klasa [2, str. 401].



Slika 2: Aktivacijske funkcije *sigmoid* i *ReLU*; preuzeto iz [2, str. 401]

Alternativno, često korištena aktivacijska funkcija u neuronskim mrežama je *ReLU* (eng. *Rectified Linear Unit*), koja je definirana kao:

$$g(z) = \text{ReLU}(z) = \max(0, z)$$

ReLU funkcija, kao na slici 2, postavlja sve negativne vrijednosti na nulu, dok zadržava sve pozitivne vrijednosti. Ova funkcija [3, str. 272] se široko koristi u dubokim neuronskim mrežama zbog svoje jednostavnosti i učinkovitosti. Ona omogućava mreži da uči brže i sprječava problem zasićenja, koji se često pojavljuje kod funkcija poput sigmoidne [2, str. 401]. Za treniranje mreže koristi se metoda unazadnog širenja gradijenta (*backpropagation*), koja minimizira funkciju gubitka, obično koristeći metodu stohastičkog gradijentnog spusta (*stochastic gradient descent*) [2, str. 429]. Funkcija gubitka može biti izražena kao kvadratna pogreška:

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

gdje je y_i stvarna vrijednost za uzorak i, a $\hat{y}_i$ predviđena vrijednost mreže. Cilj treniranja je minimizirati ovu funkciju gubitka prilagođavanjem težina i pristranosti u mreži [2, str. 402]. Nakon što mreža prođe kroz proces treniranja, može se koristiti za predviđanje na nepoznatim podacima. Predikcija se temelji na istim operacijama koje se koriste tijekom učenja, s time da su težine već optimizirane. Jednostavna neuronska mreža s jednim skrivenim slojem može se koristiti za rješavanje raznih problema, uključujući klasifikacijske i regresijske zadatke. Međutim, za složenije zadatke, kao što je prepoznavanje obrazaca u slikama, potrebne su dublje mreže, kao što su konvolucijske neuronske mreže, koje su sposobne učiti hijerarhijske značajke iz podataka [1, str. 336].

S obzirom na to da je objašnjen pojam jednostavne neuronske mreže, u nastavku se može objasniti kako je građena arhitektura CNN-a. Središnji koncept CNN-a je konvolucijski sloj, koji primjenjuje operaciju konvolucije na ulazne podatke. Konvolucija je matematička operacija koja kombinira dva skupa podataka kako bi se dobio treći. U slučaju slike, konvolucija se primjenjuje tako da se mali filtar (poznat i kao jezgra) pomiče preko slike te se za svaki položaj računa skalarni produkt između elemenata filtra i odgovarajućeg dijela slike [1, str. 332]. Matematički, operacija konvolucije na dvodimenzionalnim podacima I i filtru K definira se kao:

$$S(i, j) = (I * K)(i, j) = \sum_{m} \sum_{n} I(i - m, j - n) \cdot K(m, n)$$

gdje je $S(i, j)$ rezultat konvolucije na poziciji (i, j), I je ulazni podatak (npr. slika), a K je konvolucijski filtar [1, str. 333]. Ovaj proces omogućava detekciju lokalnih značajki kao što su rubovi, teksture i drugi složeni uzorci unutar slike [1, str. 333].

Višeslojna arhitektura CNN-a koristi više konvolucijskih slojeva, pri čemu svaki sloj može imati više filtara, što omogućava mreži da uči različite razine apstrakcije [1, str. 334]. U matematičkom smislu, ako I predstavlja ulaz s C_{in} kanala (npr. RGB slike imaju 3 kanala), izlaz iz konvolucijskog sloja s C_{out} filtara može se zapisati kao:

$$S_k(i,j) = \sigma\left(\sum_{c=1}^{C_{\text{in}}} \sum_m \sum_n I_c(i-m, j-n) \cdot K_{kc}(m,n) + b_k\right)$$

gdje je $S_k(i,j)$ karta značajki za k-ti izlazni kanal, K_{kc} su težine k-tog filtra za c-ti ulazni kanal, b_k je pristranost (bias) za k-ti kanal, a σ je aktivacijska funkcija koja se obično koristi za uvođenje nelinearnosti [1, str. 333–334], [3, str. 291–295].

Nakon što se izvede operacija konvolucije, rezultirajući signal prolazi kroz aktivacijsku funkciju koja u mrežu uvodi nelinearnost. Najčešće korištena aktivacijska funkcija u CNN-ovima je *ReLU* (eng. *Rectified Linear Unit*) koja je već ranije spomenuta. U nekim slojevima, umjesto *ReLU*, može se koristiti *Leaky ReLU*, koja je definirana kao:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{ako } x > 0 \\ \alpha x & \text{inače} \end{cases}$$

gdje je α mala pozitivna konstanta (npr. $\alpha = 0.01$). Ova funkcija omogućava da mreža zadrži male negativne vrijednosti, što može poboljšati učenje u određenim situacijama [3, str. 273].

Pooling slojevi koriste se za smanjenje dimenzionalnosti izlaza iz konvolucijskih slojeva, čime se smanjuje broj parametara i računalna složenost modela, a istovremeno zadržava informacija o najvažnijim značajkama [1, str. 340]. Najčešće korišteni pooling slojevi su *maksimalno pooling* i *prosječno pooling*. Maksimalno pooling (*max pooling*) izračunava maksimalnu vrijednost unutar malog pravokutnog prozora koji se pomiče po karti značajki:

$$P(i,j) = \max_{(m,n) \in R(i,j)} S(m,n)$$

gdje $R(i,j)$ predstavlja prozor dimenzija $p \times p$ (npr. 2×2) s centrom u točki (i,j) [4]. Ova operacija osigurava da se u svakoj podregiji sačuva najrelevantnija informacija, što dodatno povećava robusnost mreže prema malim promjenama u ulaznim podacima [1, str. 344–345].

Na kraju mreže, nalazi se jedan ili više potpuno povezanih slojeva (*fully connected layers*), gdje je svaki neuron povezan sa svim neuronima iz prethodnog sloja [3, str. 271–272]. Ovi slojevi imaju ključnu ulogu u donošenju konačnih odluka o klasifikaciji na temelju značajki koje su izdvojene u prethodnim slojevima. Izlaz svakog neurona u potpuno povezanom sloju može se matematički opisati kao:

$$y_i = \sigma\left(\sum_{j=1}^{n} W_{ij} x_j + b_i\right)$$

gdje je y_i izlaz i-tog neurona, x_j su ulazne vrijednosti, W_{ij} su težine povezane s ulazom j i neuronom i, b_i je pristranost, a σ aktivacijska funkcija. Potpuno povezani slojevi omogućuju kombiniranje i interpretaciju značajki u svrhu klasifikacije [3, str. 285].

U modernim CNN-ovima često se koriste slojevi za normalizaciju, kao što je *batch normalization*, kako bi se ubrzao proces treniranja i poboljšala stabilnost modela [5]. *Batch normalization* normalizira ulazne podatke svakog sloja tako da imaju srednju vrijednost nula i standardnu devijaciju jedan [3, str. 277]. Matematički, normalizacija unutar sloja može se izraziti kao:

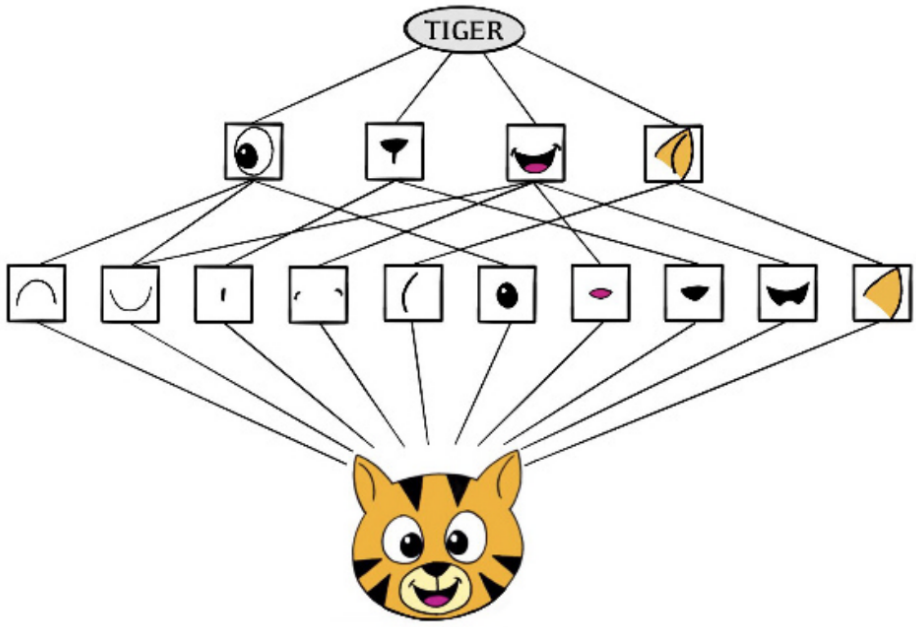
$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

gdje je x ulazni podatak, μ srednja vrijednost, σ^2 varijanca, a ϵ mala konstanta koja sprječava dijeljenje s nulom. Normalizacija pomaže u smanjenju problema eksponencijalnog rasta gradijenata tijekom propagacije kroz mrežu te omogućava brže i stabilnije učenje [3, str. 277].

Dakle, arhitektura CNN-a omogućava hijerarhijsko učenje značajki iz ulaznih podataka, počevši od jednostavnih rubova u prvim slojevima, pa sve do složenih struktura i objekata u višim slojevima [1, str. 336]. Kroz slojeve konvolucije, *pooling*, normalizacije i potpuno povezane slojeve, CNN je u stanju izvući ključne informacije i koristiti ih za učinkovitu klasifikaciju ili druge zadatke računalnog vida. Ova struktura omogućava prilagodljivost i efikasnost CNN-a u širokom rasponu primjena, što ga čini jednim od najmoćnijih alata u domeni dubokog učenja.

2.1.2. Funkcioniranje konvolucijskih slojeva

Konvolucijski slojevi su ključni elementi CNN-a koji omogućuju učinkovito izdvajanje značajki potrebnih za složene zadatke poput detekcije lica, predikcije spola i procjene dobi. U ovoj sekciji istražit će se kako konvolucijski slojevi funkcioniraju u kontekstu ovih zadataka, s naglaskom na praktične primjene i specifične izazove u obradi vizualnih podataka.



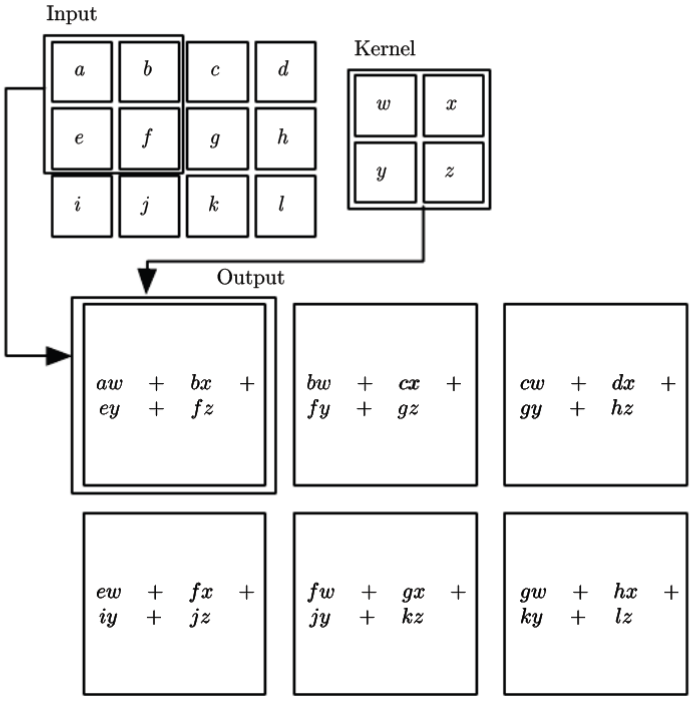
Slika 3: Shematski prikaz kako CNN klasificira sliku tigra; preuzeto iz [2, str. 407]

Na slici 3 mreža prima sliku i prepoznaje lokalne značajke. Zatim kombinira lokalne značajke kako bi stvorila složene značajke, koje u ovom primjeru uključuju oči i uši. Te složene značajke koriste se za izlaz oznake "tigar". Slično tome, u detekciji lica, mreža identificira osnovne značajke poput rubova očiju, nosa i usta te ih kombinira kako bi prepoznala cijelo lice.

Za zadatke poput detekcije lica i predikcije spola te dobi, ulazni podaci su često slike u boji koje se mogu matematički predstaviti kao višedimenzionalni tenzori. Za obojene slike, tenzor I sadrži tri dimenzije: visinu H, širinu W i broj kanala C (npr. RGB kanali). Matematički [1, str. 31–34], slika može biti izražena kao:

$$I \in \mathbb{R}^{H \times W \times C}$$

gdje je svaki element $I(h, w, c)$ intenzitet piksela na poziciji (h, w) u kanalu c. Ova reprezentacija omogućuje modelu da analizira slike kroz slojevitu strukturu, izdvajajući značajke potrebne za prepoznavanje lica, spola i dobi.

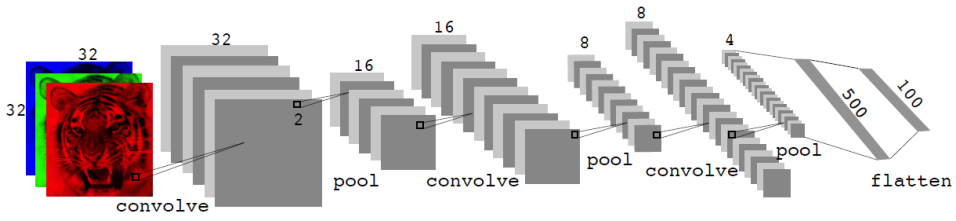


Slika 4: Primjer 2D konvolucije bez okretanja jezgre; preuzeto iz [1, str. 335]

Detekcija lica jedan je od najvažnijih zadataka u računalnom vidu, gdje se CNN-ovi pokazali izrazito učinkovitim. Na početku mreže, konvolucijski slojevi koriste male filtre za prepoznavanje osnovnih obrazaca, poput rubova i tekstura. Ti se osnovni obrasci kasnije kombiniraju u dubljim slojevima kako bi se prepoznale složenije strukture, poput očiju, nosa i usta. Matematički, konvolucija slike I s filtrom K definira se kao:

$$S(i, j, k) = \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{n=1}^{N} I(i + m, j + n, c) \cdot K(m, n, c, k) + b_k$$

gdje je $S(i, j, k)$ izlazna vrijednost na poziciji (i, j) u k-tom filtru, $M \times N$ dimenzije filtra K, C broj ulaznih kanala, a b_k pristranost za k-ti filtar [1, str. 334]. Ova operacija omogućava mreži da detektira različite značajke poput rubova, tekstura i oblika na različitim dijelovima slike. Na primjer, kod detekcije lica, konvolucijski slojevi mogu prepoznati specifične strukture koje su ključne za identifikaciju lica, kao što su oči, nos i usta.



Slika 5: Arhitektura CNN za zadatak klasifikacije; preuzeto iz [2, str. 411]

U nekim implementacijama, kao što je metoda višeslojne detekcije lica (*Multi-task Cascaded Convolutional Networks*, MTCNN) [6], mreža koristi kaskadu konvolucijskih slojeva kako bi identificirala ključne točke na licu (npr. oči, nos, usta) i precizno detektirala lice čak i u složenim scenarijima gdje lice nije u idealnoj orijentaciji ili je djelomično prekriveno.

Za zadatke predikcije spola i procjene dobi, konvolucijski slojevi kao na slici 5 koriste se za izdvajanje značajki koje su relevantne za određivanje tih atributa. Primjerice, mreža može prepoznati obrasce koji su povezani s muškim ili ženskim licima, poput linije čeljusti, oblika očiju ili strukture kostiju lica.

Nakon što se značajke izdvoje kroz konvolucijske slojeve, sljedeći slojevi (poput potpuno povezanih slojeva) koriste te značajke za donošenje konačne odluke. Na primjer, u zadatku predikcije spola, nakon što su značajke izdvojene, izlazni sloj može koristiti softmax aktivacijsku funkciju kako bi klasificirao lice kao muško ili žensko [7]. Slično tome, za procjenu dobi, izlaz može biti diskretna vrijednost koja predstavlja predviđenu dob, ili regresijski izlaz koji daje točniju procjenu u godinama.

U kontekstu detekcije lica, predikcije spola i dobi, često se koriste tehnike *data augmentation* kako bi se povećala raznolikost ulaznih podataka. Ove tehnike uključuju horizontalno i vertikalno ogledanje, rotaciju, skaliranje i translaciju slika. Na primjer, u zadatku detekcije lica, horizontalno ogledanje slike omogućava mreži da nauči prepoznati lice bez obzira na smjer u kojem je okrenuto [8]:

- **horizontalno i vertikalno ogledanje**: povećava robusnost modela prema promjenama orijentacije lica,
- **rotacija**: omogućava modelu da prepozna lice iz različitih kutova,
- **skaliranje**: pomaže u detekciji lica različitih veličina,
- **translacija**: pomiče sliku po osi x ili y, što omogućava modelu da prepozna lice bez obzira na njegovu poziciju u slici,

- **promjena intenziteta boja**: povećava robusnost prema varijacijama u osvjetljenju [2, str. 411–412].

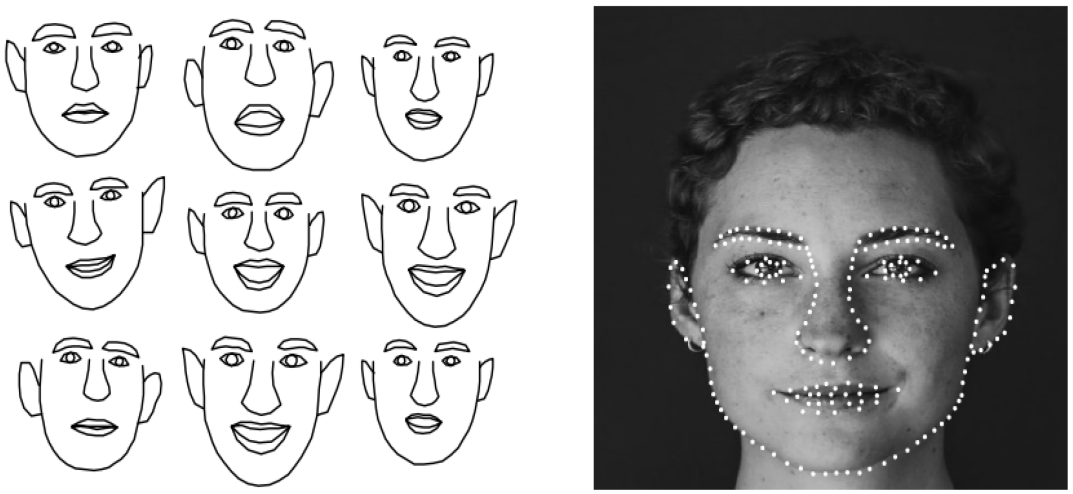
Matematički, neka transformacija T primijenjena na sliku I rezultira novom slikom $\hat{I}$:

$$\hat{I} = T(I)$$

Ove transformacije se često primjenjuju nasumično tijekom treniranja kako bi se generirale različite varijante ulaznih podataka, čime se smanjuje rizik od prekomjernog prilagođavanja modela [1, str. 345].

Funkcioniranje konvolucijskih slojeva u detekciji lica, predikciji spola i dobi odražava složenost tih zadataka i potrebu za preciznim izdvajanjem značajki. Na primjer, u stvarnim primjenama, poput sigurnosnih sustava ili analize podataka u društvenim mrežama, točnost ovih zadataka može biti presudna. Detekcija lica omogućava prepoznavanje i autentifikaciju korisnika kao na slici 6, dok predikcija spola i dobi omogućava personalizaciju korisničkog iskustva i analizu demografskih podataka. Implementacija ovih sustava često uključuje prethodno trenirane mreže, poput VGG-Face [9], koje su trenirane na velikim skupovima podataka i prilagođene specifičnim zadacima poput detekcije lica, predikcije spola i dobi.

Stoga, konvolucijski slojevi igraju ključnu ulogu u uspješnoj primjeni CNN-a u zadacima računalnog vida, kao što su detekcija lica, predikcija spola i dobi. Kroz kombinaciju različitih tehnika, kao što su konvolucija, pooling, *data augmentation* i optimizacija parametara, moguće je postići visoku preciznost i robusnost u ovim zadacima, čime se CNN-ovi potvrđuju kao nezamjenjivi alati u suvremenoj analizi vizualnih podataka [1, str. 372].



Slika 6: Globalni oblici lica: (a) tipični trening globalnih oblika lica koji se sastoje od crta lica, kao što su oči, usta, nos, obrve i uši, (b) točke modela projicirane na sliku za vježbanje s licem koje proizvodi globalne oblike lica; preuzeto iz [10]

2.2. Stabilnost i robusnost CNN-a

Stabilnost i robusnost CNN ključni su za pouzdanost modela u realnim primjenama, kao što su prepoznavanje lica, klasifikacija spola i dobi. CNN-i su često izloženi raznim vrstama ulaznih promjena i šuma koji mogu značajno utjecati na performanse modela. U tom kontekstu, stabilnost se odnosi na sposobnost mreže da održava konzistentne performanse unatoč malim promjenama u ulazu, dok robusnost mjeri otpornost modela na veće, često namjerne promjene, kao što su *adversarial* napadi [11]. S obzirom na široku primjenu CNN-a u zadacima računalnog vida, osiguravanje stabilnosti i robusnosti postaje presudno za njihovu primjenu u svakodnevnom životu, npr. u sigurnosnim sustavima baziranim na prepoznavanju lica.

2.2.1. Definicija stabilnosti

Stabilnost CNN od suštinske je važnosti za osiguranje konzistentnih izlaznih rezultata, unatoč prisutnosti malih promjena u ulaznim podacima. U dubokom učenju, stabilnost se odnosi na sposobnost mreže da proizvodi konzistentne predikcije kada su ulazni podaci slični, čak i uz prisutnost malih perturbacija. Ova je karakteristika posebno važna u zadacima kao što su prepoznavanje lica, spola i dobi, gdje varijacije u uvjetima snimanja, poput promjena u osvjetljenju, rotaciji ili šumu, ne bi trebale rezultirati značajnim odstupanjima u izlazima mreže [12], [13].

Matematička formalizacija stabilnosti u neuronskim mrežama često se ostvaruje korištenjem koncepta Lipschitzove konstantne. Lipschitzova stabilnost osigurava da male promjene u ulazu ne uzrokuju neproporcionalno velike promjene u izlazima modela. Lipschitzova konstanta L [14] definira se kao minimalna konstanta koja ograničava omjer promjene u izlazu u odnosu na promjenu u ulazu, prema sljedećoj formuli:

$$\|f(I + \delta I) - f(I)\| \leq L\|\delta I\|$$

gdje je $f(I)$ funkcija koja mapira ulaznu sliku I na izlazne vrijednosti modela, a δI mala perturbacija u ulaznim podacima [11]. Manja vrijednost L znači da model reagira kontrolirano na promjene u ulazu, dok velika vrijednost L može ukazivati na preosjetljivost modela, čime se smanjuje njegova stabilnost.

U kontekstu prepoznavanja lica, stabilna mreža trebala bi prepoznati osobu ili predvidjeti spol unatoč manjim promjenama poput osvjetljenja ili rotacije lica. Na primjer, promjena kuta snimanja lica ne bi trebala drastično promijeniti izlaz mreže, već bi stabilna mreža trebala proizvesti slične predikcije za slične ulazne slike [6].

Funkcija f se smatra L-Lipschitz kontinuiranom ako postoji konstanta L takva da za sve ulaze x i y vrijedi:

$$\|f(x) - f(y)\| \leq L\|x - y\|$$

Ovaj koncept osigurava da razlika u izlazima funkcije bude proporcionalna razlici u ulazima, što omogućava kontroliranu reakciju mreže na promjene ulaza [11]. U neuronskim mrežama, Lipschitzova konstanta može se tumačiti kroz težine modela: ako su težine prevelike, i male promjene u ulazima mogu uzrokovati velike promjene u izlazima, čineći model nestabilnim.

Regularizacijske metode [15], kao što je L2 regularizacija, igraju ključnu ulogu u održavanju stabilnosti modela ograničavanjem veličine težina modela. L2 regularizacija smanjuje prekomjerno prilagođavanje modela na specifične ulazne uzorke, pomažući u kontroliranju L-Lipschitz konstante modela:

$$L_{\text{total}} = L_{\text{data}} + \lambda \sum_{i=1}^{n} w_i^2$$

Ovaj izraz dodaje kaznu kvadrata težina modela na funkciju gubitka, čime se penaliziraju velike težine i postiže stabilniji model [16]. U praksi, ova tehnika smanjuje Lipschitzovu konstantu, što dovodi do bolje otpornosti mreže na male promjene u ulaznim podacima. Primjerice, u prepoznavanju lica, L2 regularizacija može pomoći u postizanju točnijih rezultata, čak i kada slike variraju u kvaliteti ili osvjetljenju.

U praktičnim primjenama, poput prepoznavanja lica, spola i dobi, stabilnost mreže osigurava da model daje konzistentne rezultate unatoč varijacijama u uvjetima snimanja. Na primjer, stabilna mreža trebala bi točno prepoznati spol osobe bez obzira na promjene u osvjetljenju ili kutu snimanja [7]. Održavanje stabilnosti u ovakvim zadacima je ključno kako bi se osiguralo da model bude upotrebljiv u realnim scenarijima, kao što su biometrijski sustavi ili sigurnosni nadzori.

Primjenom Lipschitzove stabilnosti, možemo osigurati da male promjene u ulaznim podacima, poput blage rotacije lica ili promjene osvjetljenja, ne uzrokuju velike promjene u predikcijama mreže. Na primjer, u zadatku prepoznavanja spola, mreža koja je stabilna prema Lipschitzovom kriteriju trebala bi dosljedno predviđati spol osobe čak i u prisutnosti šuma ili perturbacija [6].

Jedna od glavnih tehnika za povećanje stabilnosti CNN-a je *batch normalization*. *Batch normalization* smanjuje unutarnje oscilacije unutar mreže normaliziranjem distribucija podataka svakog sloja, što stabilizira treniranje i poboljšava ponašanje modela [5]. Ova tehnika pomaže mreži da bude stabilnija prema perturbacijama ulaza, jer smanjuje varijacije unutar slojeva mreže. Uz *batch normalization*, tehnika *dropout* također povećava stabilnost mreže nasumičnim isključivanjem određenih neuronskih jedinica tijekom treniranja, čime se sprječava da model postane previše ovisan o određenim ulaznim značajkama [17]. Ova metoda trenira mrežu da bude robusnija, jer potiče generalizaciju.

Jedan od ključnih izazova za stabilnost CNN-a su *adversarial* primjeri – ulazi s malim, jedva primjetnim promjenama koje mogu značajno promijeniti izlaz mreže [18]. Primjeri pokazuju koliko je stabilnost važna u osiguravanju robusnosti mreže. Na primjer, mala promjena u pikselima slike može prevariti mrežu da pogrešno prepozna lice ili spol osobe [19]. Kako bi

se mreža zaštitila od ovakvih napada, koristi se *adversarial* treniranje [18], [20], gdje se model trenira na ovim primjerima kako bi postao otporniji. Ova tehnika poboljšava stabilnost modela, osiguravajući da on ne postane ranjiv na male manipulacije ulazima.

Stabilnost CNN-a ključna je za postizanje dosljednih i točnih rezultata u složenim zadacima poput prepoznavanja lica, spola i dobi. Lipschitzova konstanta pruža matematički okvir za razumijevanje i poboljšanje stabilnosti, dok tehnike poput regularizacije, *batch normalization*-a i *dropout*-a osiguravaju dugoročnu stabilnost modela. Kombinacija ovih tehnika osigurava da CNN može izdržati perturbacije u ulaznim podacima i zadržati visoku razinu performansi [6], [16], [19].

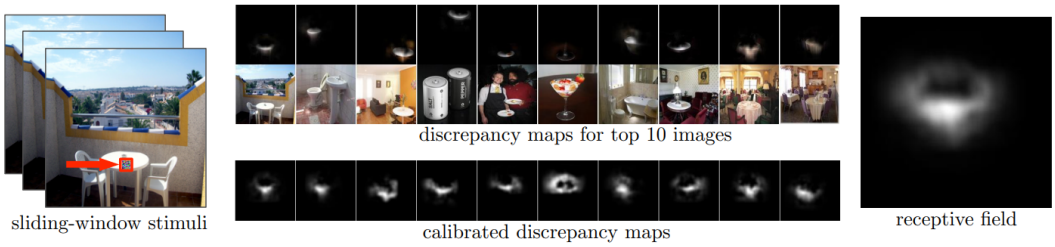
2.2.2. Faktori koji utječu na stabilnost

Stabilnost CNN-a ovisi o nekoliko ključnih faktora koji proizlaze iz arhitektonskih i algoritamskih odluka. Ti faktori mogu značajno promijeniti kako model obrađuje podatke te kako odgovara na perturbacije ili promjene u ulazima. U ovoj podsekciji će se detaljnije razmotriti kako različiti faktori utječu na stabilnost CNN-a u zadacima poput prepoznavanja lica te predikcije spola i dobi.

Veličina filtara je jedan od osnovnih faktora koji direktno utječe na performanse CNN-a, posebice na njegovu stabilnost. Filtri su "prozori" kroz koje mreža gleda ulazne podatke i iz njih izvlači značajke. Veći filtri hvataju šire prostorne odnose, dok manji filtri detektiraju fine detalje unutar slike. U praksi, manji filtri poput 3×3 ili 5×5 osiguravaju veću osjetljivost prema malim značajkama, kao što su oči, nos ili usta u zadacima prepoznavanja lica [21]. Međutim, zbog njihove preciznosti, manji filtri također postaju osjetljiviji na šum u podacima, što može negativno utjecati na stabilnost mreže. Na primjer, mala promjena u osvjetljenju ili šum može utjecati na način na koji mali filtar detektira fine značajke, što može uzrokovati varijabilne rezultate. S druge strane, veći filtri, poput 7×7 ili 11×11, obuhvaćaju šire prostore slike i općenito su otporniji na šum jer ne ovise o sitnim detaljima. Takvi filtri poboljšavaju generalizaciju mreže jer hvataju šire prostorne odnose između različitih dijelova slike. Međutim, zbog svog obuhvata, oni mogu izgubiti na preciznosti kada je potrebno prepoznati male, ali bitne značajke, poput detalja na licu. U zadatku prepoznavanja lica, stabilnost CNN-a ovisi o odgovarajućoj veličini filtara jer lice sadrži mnoge fine značajke koje treba točno detektirati, ali je također potrebno da mreža bude otporna na šum i perturbacije. Na primjer, prepoznavanje identiteta osobe oslanja se na stabilnost mreže pri prepoznavanju specifičnih značajki lica (npr. oblik očiju ili usta), čak i kada je slika izložena malim promjenama u osvjetljenju ili kutu snimanja.

Osim veličine filtara, važan faktor koji utječe na stabilnost CNN-a je i receptivno polje. Receptivno polje označava dio slike na koji jedan neuron odgovara. Što je mreža dublja i složenija, to veće receptivno polje može postati, što omogućava mreži da izvuče više informacija iz šireg konteksta slike. Manja receptivna polja su korisna za hvatanje lokalnih značajki, poput rubova i tekstura, dok veća receptivna polja kao primjer sa slike 7 omogućuju mreži da prepoznaje globalne obrasce i objekte na slici [22]. U zadatku prepoznavanja lica, balansiranje između lokalnih i globalnih značajki je ključno za postizanje stabilnog modela. Na primjer, stabilnost mreže pri prepoznavanju lica zahtijeva da mreža ne bude previše osjetljiva na male

promjene na pojedinim dijelovima lica, ali da bude sposobna prepoznati cjelokupnu strukturu lica kao entitet, što osigurava dosljednost u rezultatima.



Slika 7: Cjevovod za procjenu receptivnog polja svake jedinice. Svaki podražaj u kliznom prozoru sadrži mali nasumični dio (primjer označen crvenom strelicom) na različitim prostornim lokacijama. Uspoređujući aktivacijski odgovor podražaja kliznog prozora s aktivacijskim odgovorom izvorne slike, dobiva se mapa odstupanja za svaku sliku (sredina gore). Zbrajanjem kalibriranih mapa odstupanja (sredina dolje) za najbolje rangirane slike, dobiva se stvarno receptivno polje te jedinice (desno); preuzeto iz [22]

Normalizacija podataka je još jedan važan čimbenik koji utječe na stabilnost CNN-a. *Batch normalization* je jedna od tehnika koja smanjuje unutarnju varijaciju unutar mreže i omogućuje stabilnije treniranje. Ona radi tako da normalizira izlaze svakog sloja mreže na nultu srednju vrijednost i standardnu devijaciju jedan, što smanjuje varijaciju između *batch*-eva i ubrzava konvergenciju [5]. *Batch normalization* također pomaže u smanjenju osjetljivosti modela na promjene u ulazima jer stabilizira distribuciju podataka unutar mreže. U zadacima prepoznavanja lica, to znači da CNN može biti stabilniji prema promjenama u osvjetljenju ili kutu lica jer *batch normalization* osigurava konzistentnost između različitih prikaza lica unutar *batch*-eva. Na primjer, u zadacima gdje se treniraju CNN modeli za prepoznavanje spola ili dobi, normalizacija osigurava da male promjene u uvjetima snimanja lica ne uzrokuju dramatične promjene u izlazima. To omogućava stabilniju i robusniju mrežu koja može generalizirati na nove slike u stvarnim uvjetima snimanja.

Dropout je tehnika koja se koristi za poboljšanje stabilnosti i robusnosti mreža tijekom treniranja. Ova tehnika nasumično isključuje određeni postotak neuronskih jedinica u slojevima tijekom svakog koraka treniranja, čime sprječava mrežu da postane previše ovisna o određenim značajkama [17]. U osnovi, *dropout* poboljšava stabilnost modela jer trenira mrežu da ne bude ovisna o specifičnim obrascima u podacima. Time se smanjuje rizik od prekomjernog prilagođavanja (overfittinga) na trening skup podataka i omogućuje bolja generalizacija na neviđene podatke. To je posebno važno u zadacima poput prepoznavanja spola i dobi, gdje su treninzi podaci često ograničeni ili nejednoliko distribuirani. Na primjer, kod prepoznavanja dobi osobe s lica, *dropout* može spriječiti mrežu da prekomjerno nauči obrasce povezane s određenim demografskim skupinama te omogućiti stabilniju mrežu koja generalizira na širu populaciju.

Adversarial treniranje je specifična tehnika koja poboljšava stabilnost CNN-a prema namjernim napadima i neprijateljskim perturbacijama. *Adversarial* primjeri su ulazi koji su posebno dizajnirani kako bi prevarili mrežu, a često uključuju vrlo male i gotovo neprimjetne promjene koje uzrokuju značajne promjene u izlazu mreže [19]. U *adversarial* treniranju, mreža se trenira na ovim neprijateljskim primjerima kako bi postala robusnija prema takvim napadima.

Na taj način mreža uči prepoznati i ignorirati neprirodne promjene u ulazu, što poboljšava njenu stabilnost u stvarnim uvjetima rada. Na primjer, u zadatku prepoznavanja lica, *adversarial* treniranje omogućuje mreži da prepozna lice čak i ako su na slici dodane perturbacije koje pokušavaju zbuniti model. Time se poboljšava ukupna robusnost modela, što je ključno za sigurnosne sustave koji se oslanjaju na prepoznavanje lica.

Stabilnost CNN-a ovisi o više međusobno povezanih faktora, od veličine filtara do naprednih tehnika poput *batch normalization*, *dropout* i *adversarial* treniranja. Svaki od ovih faktora ima specifičan utjecaj na način na koji mreža obrađuje podatke i reagira na promjene u ulazu. U zadacima prepoznavanja lica te predikcije spola i dobi, stabilna mreža može značiti razliku između preciznih i konzistentnih predikcija te rezultata koji su osjetljivi na male promjene u ulaznim podacima. Korištenjem odgovarajućih arhitektonskih i algoritamskih rješenja, moguće je osigurati stabilnost CNN-a čak i u prisutnosti perturbacija ili neprijateljskih napada.

2.2.3. Utjecaj promjena na performanse

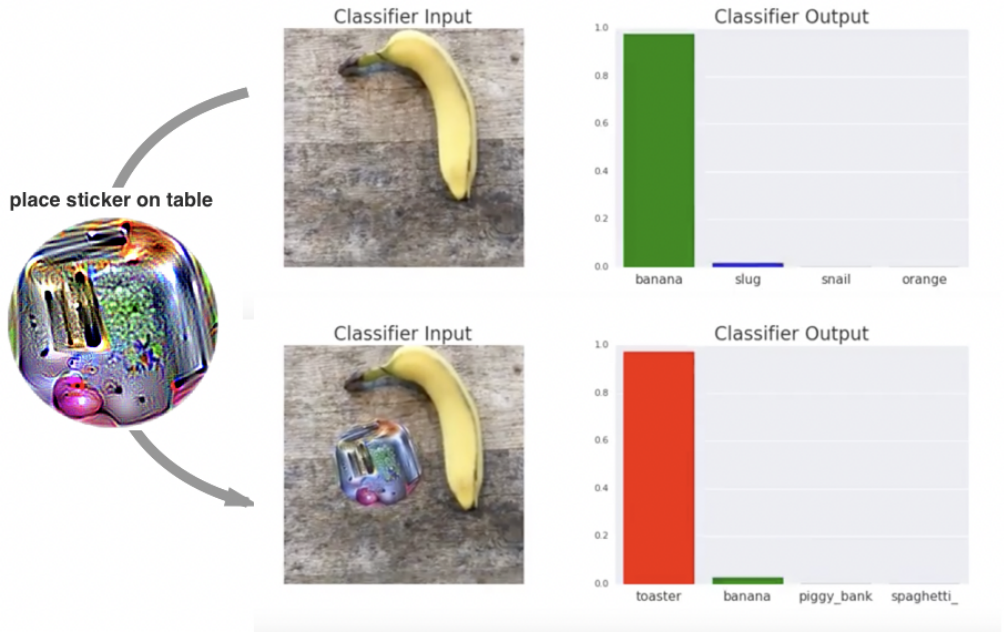
Promjene u ulaznim podacima mogu značajno utjecati na performanse CNN. Iako su CNN-ovi dizajnirani da budu robusni i otporni na varijacije, njihove performanse mogu oslabiti ako su te promjene prevelike ili previše učestale.

Jedan od najčešćih problema u zadacima računalnog vida, poput prepoznavanja lica, je varijacija u uvjetima osvjetljenja. Promjene u intenzitetu ili boji svjetla mogu uzrokovati da CNN-i ne prepoznaju pravilno obrasce i značajke na slici [7]. Promjene u osvjetljenju mogu uzrokovati varijacije u kontrastu i sjenama na slici, što rezultira smanjenjem točnosti u zadacima prepoznavanja lica ili predikciji spola i dobi. CNN-ovi, osobito oni trenirani na jednoličnim skupovima slika s minimalnim varijacijama u osvjetljenju, mogu biti posebno osjetljivi na promjene svjetline jer će mreža drugačije prepoznati rubove i teksture. Na primjer, lice snimljeno pod jakim svjetlom može izgledati potpuno drugačije nego lice snimljeno pod slabim ili difuznim svjetlom, što može uzrokovati da mreža pogrešno klasificira identitet ili spol. CNN se oslanja na rubove i teksture unutar slika kako bi prepoznao ključne značajke lica. Međutim, osvjetljenje može "sakriti" ili promijeniti te značajke, zbunjujući mrežu u prepoznavanju lica ili u donošenju ispravnih predikcija. Korištenje tehnika kao što su *data augmentation* može donekle smanjiti ovaj problem, ali ako promjene u osvjetljenju nisu uključene u trening skup, mreža će biti ranjiva na ovaj tip promjena.

Promjene u orijentaciji (rotacija) i poziciji lica na slici mogu također značajno utjecati na performanse CNN-a. Lice prikazano pod različitim kutom može biti teško prepoznati jer neki dijelovi lica mogu biti izvan vidokruga ili zaklonjeni. Promjena u kutu snimanja lica može učiniti da mreža zaboravi ili ne prepozna bitne značajke. To posebno dolazi do izražaja u zadacima poput prepoznavanja identiteta ili predikcije spola, gdje mreža može dati različite rezultate ovisno o tome iz kojeg je kuta lice snimljeno. Na primjer, lice okrenuto prema kameri mreža može prepoznati, ali lice okrenuto pod kutom od 45 stupnjeva može se prepoznati s puno nižom točnošću. CNN-ovi nisu inherentno rotacijski ili translacijski invarijantni, što znači da različiti kutovi i pozicije mogu dovesti do različitih aktivacija u konvolucijskim slojevima. Bez dodatnih tehnika kao što su *spatial pooling* ili *data augmentation*, mreža može biti osjetljiva na

ove varijacije, što dovodi do značajnog smanjenja performansi u stvarnim uvjetima, gdje lica rijetko ostaju u istom položaju.

Šum, bilo da je to nasumičan šum uzrokovan lošom kvalitetom slike ili šum koji je namjerno uveden kao dio *adversarial* napada, može dramatično utjecati na performanse CNN-a [19]. Dodavanje šuma, osobito visokofrekvencijskog šuma, može uzrokovati pogrešne predikcije jer CNN može početi prepoznavati šum kao značajke koje nisu stvarno prisutne u slici. To može uzrokovati smanjenje točnosti u zadacima prepoznavanja lica i predikcije spola i dobi, osobito kada je šum prisutan u područjima ključnih značajki, poput očiju ili nosa. CNN-ovi su osjetljivi na šum jer šum može biti "pročitana" kao ključna značajka, osobito ako se nalazi u regijama gdje mreža očekuje korisne informacije. Na primjer, u zadatku prepoznavanja lica, šum u regiji oko očiju može ometati mrežu u ispravnom prepoznavanju lica. *Adversarial* treniranje može donekle pomoći u zaštiti modela od ovog tipa napada, ali osjetljivost na šum ostaje veliki izazov za mnoge mreže.



Slika 8: Dodavanje male mrlje na sliku zbunjuje CNN te klasificira bananu kao toster; preuzeto iz [23]

Adversarial primjeri kao na slici 8 predstavljaju sofisticirane promjene u ulazu koje su dizajnirane da obmanu mrežu i prouzrokuju pogrešne predikcije. To mogu biti male, jedva vidljive perturbacije koje drastično mijenjaju izlaz mreže. *Adversarial* primjeri mogu uzrokovati značajan pad performansi, jer mreža može potpuno pogrešno klasificirati ulazne slike. U zadacima poput prepoznavanja lica ili predikcije spola, mala promjena u pikselima slike može uzrokovati da mreža krivo identificira osobu ili spol. CNN-ovi nisu dizajnirani da razlikuju namjerno uvedene male promjene (*adversarial* primjere) od prirodnih promjena u ulazu. *Adversarial* treniranje, gdje se mreža trenira na otkrivanje i otpor prema tim promjenama, može značajno smanjiti utjecaj ovih napada, ali ostaje činjenica da su CNN-ovi vrlo ranjivi na ove vrste perturbacija zbog svoje osjetljivosti na male promjene u ulazu [19].

Smanjenje rezolucije slike može uzrokovati gubitak informacija u važnim značajkama, kao što su rubovi i teksture, što direktno utječe na performanse CNN-a. Slike niske rezolucije mogu uzrokovati da CNN ne detektira važne značajke lica, što rezultira pogrešnim predikcijama. U zadacima prepoznavanja lica, smanjenje rezolucije može zamagliti detalje oko očiju, nosa ili usta, što su ključne značajke u prepoznavanju identiteta ili spola. CNN-ovi se oslanjaju na visoku kvalitetu slika kako bi ispravno prepoznali značajke. Kada rezolucija padne, rubovi i detalji postaju mutni, što otežava mreži da točno identificira lica ili prepozna dob i spol. *Super-resolution* tehnike mogu pomoći u rješavanju ovog problema, ali mreža i dalje treba biti trenirana na slikama s različitim rezolucijama kako bi se osigurala robusnost.

Ako dijelovi slike nedostaju ili su obrezani, CNN možda neće moći detektirati sve potrebne informacije za ispravnu klasifikaciju. Obrezivanje slika tako da nedostaju dijelovi lica (npr. ako je čelo ili dio brade izvan slike) može smanjiti točnost modela jer mreža možda neće imati dovoljno informacija za donošenje ispravne odluke. CNN se oslanja na kompletne informacije sa slike kako bi izgradio predikciju. Ako ključni dijelovi lica nedostaju, mreža može pogrešno zaključiti o identitetu, spolu ili dobi osobe. Tehnike poput *data augmentation*-a koje uključuju različite razine obrezivanja slike mogu pomoći u smanjenju ovog efekta, ali mreža i dalje može imati poteškoće u prepoznavanju lica kada su važni dijelovi izvan okvira.

Razne promjene u ulaznim podacima, bilo da su prirodne (promjene u osvjetljenju, kutu snimanja) ili namjerno uvedene (*adversarial* primjeri, šum), mogu značajno narušiti performanse CNN-a. Dok postoje metode poput *data augmentation* i *adversarial* treniranja za povećanje otpornosti mreže na ove promjene, one još uvijek predstavljaju izazove u zadacima poput prepoznavanja lica i predikcije spola i dobi. Razumijevanje učinka ovih promjena ključno je za razvoj stabilnih i robusnih CNN modela koji mogu raditi u realnim uvjetima.

3. Analiza skupova podataka

U ovom poglavlju analizirat će se skupovi podataka koji se koriste za treniranje i evaluaciju modela za prepoznavanje lica, predikciju spola i dobi. Poseban fokus bit će stavljen na IMDB-Wiki skup podataka, jedan od najopsežnijih javno dostupnih skupova slika lica, koji se koristi za treniranje modela dubokog učenja u zadacima računalne vizije. Kroz nekoliko sekcija, detaljno će se obraditi proces pripreme podataka, uključujući procese čišćenja i odabira značajki za treniranje modela.

Skupovi podataka su ključni za razvoj pouzdanih i točnih modela u domeni računalnog vida. U svrhu kvalitetnog treniranja CNN-a za prepoznavanje spola i dobi, skupovi podataka moraju biti pažljivo odabrani, pripremljeni i očišćeni. Proces čišćenja podataka podrazumijeva uklanjanje nekonzistentnosti, nepotpunih ili neispravnih podataka, dok odabir značajki za treniranje osigurava da se model fokusira na relevantne informacije iz podataka.

U nastavku, prvo će se detaljno opisati odabrani skup podataka IMDB-Wiki, a potom će se u sljedećim sekcijama objasniti kako se provode procesi čišćenja i odabira značajki.

3.1. Pregled odabranih skupova podataka IMDB-Wiki

IMDB-Wiki je jedan od najopsežnijih i najkorištenijih skupova podataka u domeni računalnog vida, osobito u zadacima koji uključuju prepoznavanje spola i dobi s lica. Skup podataka je javno dostupan i predstavlja važan resurs za istraživače i stručnjake u području dubokog učenja i računalnog vida. Izvorno su ga kreirali Rasmus Rothe, Radu Timofte i Luc Van Gool sa Švicarskog federalnog instituta za tehnologiju u Zürichu (ETH Zürich), a detaljno je opisan u njihovim radovima [24], [25].

IMDB-Wiki skup podataka sa slike 9 sastoji se od dvaju glavnih podskupova: **IMDB** i **Wiki**, koji zajedno čine više od pola milijuna slika lica različitih osoba. Svaka slika u skupu anotirana je s informacijama o spolu i dobi osobe, pri čemu su podaci o dobi preuzeti iz dostupnih izvora, poput godina rođenja poznatih osoba u IMDB-u i datuma kada su fotografije postavljene na Wiki. Zbog ovih obilježja, IMDB-Wiki je postao jedan od najopsežnijih i najvažnijih skupova podataka za treniranje modela koji predviđaju dob i spol na temelju slika lica.

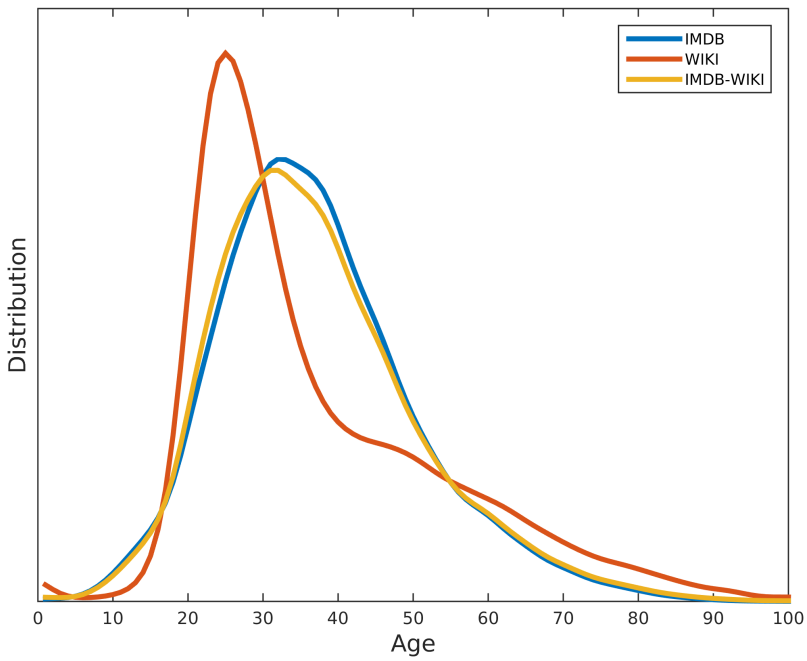
Slika 9: IMDB i Wikipedia skupovi podataka; preuzeto iz [25]

IMDB podskup sadrži 460.723 slike lica poznatih osoba iz filmske industrije, čiji su podaci preuzeti s IMDB (*Internet Movie Database*) platforme. Svaka slika je povezana s metapodacima, kao što su ime glumca, godina rođenja i spol. Ovi podaci omogućavaju točne anotacije za dob i spol svake osobe. Dob osobe je izračunata na temelju godine rođenja i datuma kada je slika snimljena ili postavljena na internet. Zbog svoje veličine i raznolikosti, IMDB podskup idealan je za treniranje modela dubokog učenja koji predviđaju dob i spol [25].

Jedan od izazova u radu s IMDB podskupom je varijabilnost u kvaliteti slika. Naime, slike su prikupljene s interneta, što znači da se mogu razlikovati u rezoluciji, osvjetljenju, ekspresijama lica i pozama. To može otežati treniranje modela, osobito ako nije provedeno temeljito čišćenje podataka. Međutim, ova varijabilnost također može biti korisna jer omogućava modelu da se prilagodi različitim uvjetima snimanja, čime postaje otporniji i robusniji u realnim primjenama.

Wiki podskup sadrži 62.328 slika lica osoba čiji su podaci preuzeti s Wikipedije. Anotacije za dob i spol također su preuzete s Wikipedije na temelju dostupnih biografskih informacija. Kao i kod IMDB podskupa, dob osobe je izračunata na temelju godine rođenja i datuma kada je slika postavljena na internet. Wiki podskup sadrži slike poznatih osoba iz različitih područja, uključujući politiku, sport, umjetnost i znanost. Ovaj podskup pruža dodatnu raznolikost u podacima, čime se omogućava bolja generalizacija modela [25].

Jedna od prednosti Wiki podskupa je to što sadrži slike osoba iz različitih demografskih skupina, uključujući različite dobi, etničke skupine i spolove. To omogućava modelima treniranim na ovom skupu da budu precizniji i generaliziraniji u zadacima predikcije spola i dobi. Međutim, kao i kod IMDB podskupa, slike iz Wiki podskupa također mogu varirati u kvaliteti, što zahtijeva temeljitu pripremu i čišćenje podataka prije treniranja modela.

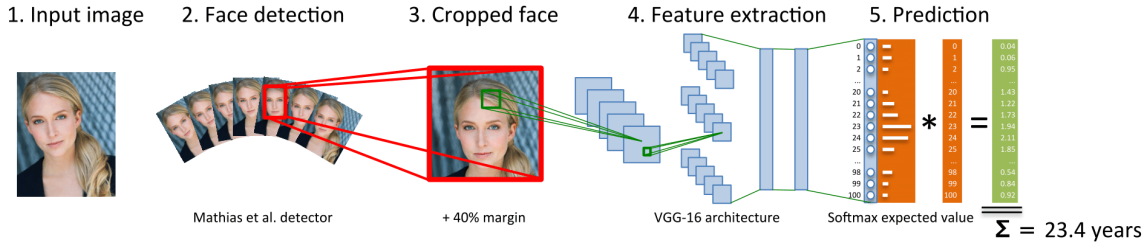


Slika 10: Distribucija podataka prema godinama; preuzeto iz [25]

Kvaliteta slika u IMDB-Wiki skupu podataka nije uvijek konzistentna. Budući da su slike prikupljene iz javnih izvora, poput IMDB-a i Wikipedije, kvaliteta slika može varirati od visoke razlučivosti do niskokvalitetnih slika s velikim količinama šuma. Nadalje, lica na slikama mogu biti djelomično zaklonjena, izobličena ili prikazana pod različitim kutovima, što dodatno otežava treniranje modela. Stoga je nužno provesti temeljito čišćenje i filtriranje podataka kako bi se uklonile slike koje bi mogle negativno utjecati na performanse modela [24].

Distribucija dobi sa slike 10 i spola u IMDB-Wiki skupu podataka također je neujednačena. Većina slika prikazuje osobe između 20 i 50 godina, dok je zastupljenost starijih osoba (iznad 60 godina) i djece (ispod 10 godina) znatno manja. Također, u skupu podataka postoji blaga prekomjerna zastupljenost muškaraca u odnosu na žene. Ove nejednakosti u distribuciji mogu utjecati na točnost modela, osobito u predikcijama spola i dobi za manje zastupljene dobne skupine ili spolove. Stoga je prilikom treniranja modela važno koristiti tehnike balansiranja podataka kako bi se osigurala ravnomjernija zastupljenost svih demografskih skupina [24].

IMDB-Wiki skup podataka koristi se prvenstveno za treniranje modela koji predviđaju dob i spol na temelju slika lica kao na slici 11. Ovi modeli imaju široku primjenu u različitim područjima, uključujući sigurnosne sustave, biometrijske aplikacije, marketing, zdravstvo i društvene medije. Na primjer, modeli trenirani na IMDB-Wiki skupu mogu se koristiti u sigurnosnim sustavima za prepoznavanje identiteta ili u marketinškim kampanjama za personalizaciju oglasa na temelju dobi i spola korisnika.



Slika 11: Proces predikcije godina neke osobe; preuzeto iz [25]

Jedan od najpoznatijih modela treniranih na IMDB-Wiki skupu podataka je **DEX *(Deep EXpectation)*** model, koji su razvili Rasmus Rothe i njegovi suradnici [25]. DEX model koristi CNN za predikciju dobi i spola s visokom preciznošću i pokazao se vrlo uspješnim na raznim benchmark testovima. Njegova točnost postignuta je zahvaljujući velikoj količini podataka i njihovoj raznolikosti, kao i korištenju naprednih tehnika dubokog učenja.

3.2. Priprema i čišćenje podataka

U ovom poglavlju detaljno se opisuju svi koraci poduzeti kako bi se IMDB-Wiki skup podataka pripremio za treniranje modela za prepoznavanje dobi i spola. S obzirom na to da kvalitetan skup podataka predstavlja temelj uspjeha modela, nužno je provesti detaljnu analizu, filtraciju, validaciju te selekciju podataka koji će biti korišteni u konačnom skupu.

3.2.1. Procesi čišćenja podataka

Proces čišćenja podataka ključan je korak u svakom projektu strojnog učenja. Ispravnost i kvaliteta podataka značajno utječu na performanse modela i točnost predikcija. U ovom radu, cilj je bio pripremiti i očistiti IMDB-Wiki skup podataka kako bi bio prikladan za treniranje modela koji predviđa dob i spol na temelju slika lica. U tu svrhu, razvijen je programski kod u Pythonu koji sadrži nekoliko međusobno povezanih klasa i metoda koje olakšavaju učitavanje, validaciju i pripremu slika za daljnje korake.

Osnovna struktura koda uključuje tri ključne komponente:

- **ImageProcessor**: klasa odgovorna za prikupljanje, validaciju i spremanje valjanih i nevaljanih slika.
- **DataExtractor**: klasa koja omogućava ekstrakciju relevantnih metapodataka iz datoteka koje sadrže IMDB-Wiki skup podataka.
- **DatasetAnalyzer**: klasa koja omogućuje analizu i vizualizaciju valjanih slika, zajedno s pripadajućim metapodacima (dob i spol).

Svi ti procesi detaljno su opisani u narednim podsekcijama.

3.2.1.1. Učitavanje i priprema slika

Prvi korak u procesu čišćenja podataka uključuje prikupljanje i spremanje svih putanja do slika koje se nalaze unutar direktorija IMDB i Wiki podskupova podataka. Kako bi se prikupljanje svih slika u ovim direktorijima moglo automatizirati, korištena je funkcija `rglob()` unutar klase `ImageProcessor`, koja omogućuje rekurzivno pretraživanje poddirektorija i pronalazak svih datoteka s ekstenzijom `.jpg`. Ova metoda osigurava učinkovito pronalaženje slika, neovisno o složenosti strukture direktorija.

Isječak koda 1: Sakupljanje svih datoteka slika

```
1         def _collect_all_filenames(self) -> list:
2             wiki_filenames = [str(f.relative_to(self.wiki_path)) for f in self.
               wiki_path.rglob('*.jpg')]
3             imdb_filenames = [str(f.relative_to(self.imdb_path)) for f in self.
               imdb_path.rglob('*.jpg')]
4             return wiki_filenames + imdb_filenames
```

Ova metoda 1 prikuplja putanje slika i pohranjuje ih kao listu, omogućujući daljnju obradu tih slika u kasnijim fazama. Važno je napomenuti da prikupljanje slika iz direktorija nije dovoljno da bi se osigurala njihova valjanost. Stoga, u sljedećem koraku, provjerava se čitljivost svake slike te je li na slici prisutno lice, budući da je model treniran za prepoznavanje dobnih i spolnih karakteristika iz slika lica.

3.2.1.2. Validacija slika

Sljedeći ključan korak u procesu čišćenja podataka jest validacija svake slike kako bi se osiguralo da model koristi samo relevantne i ispravne podatke. Ova validacija se provodi kroz dvije faze:

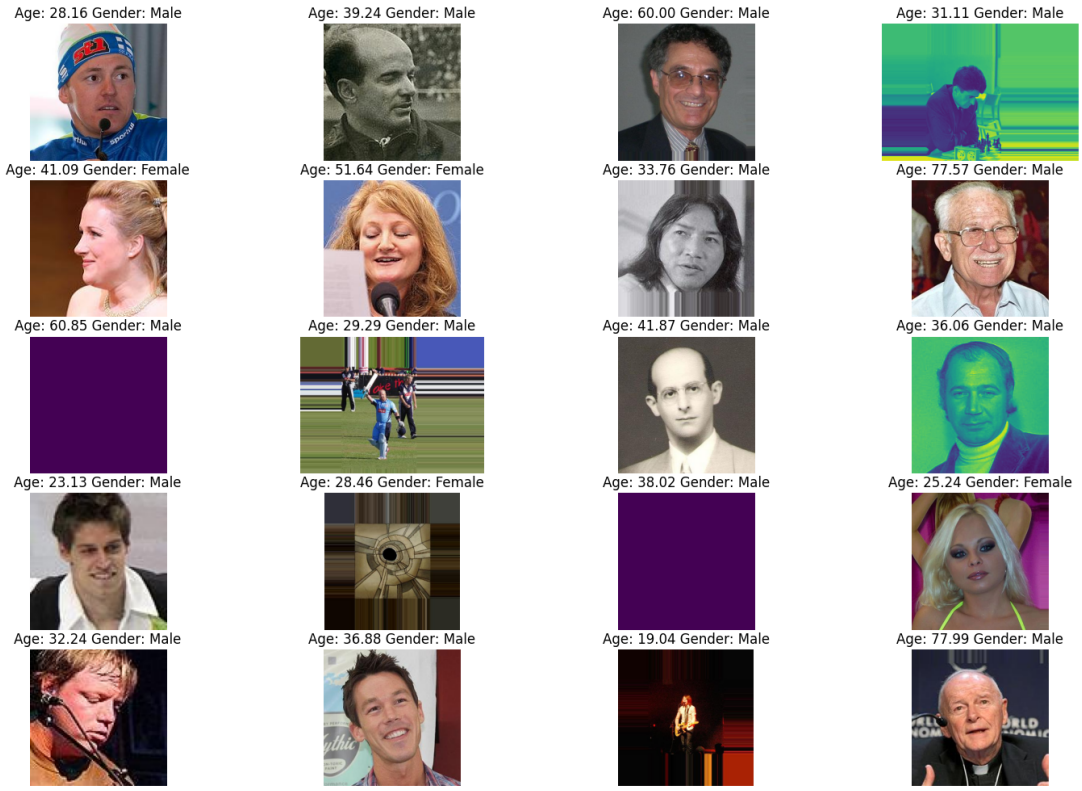
1. Provjera čitljivosti slike (npr. slika nije oštećena).
2. Provjera prisutnosti lica na slici.

Za validaciju slike razvijena je metoda `is_valid_image`, prikazana u metodi 2. Ova metoda najprije pokušava otvoriti svaku sliku korištenjem Pythonove biblioteke PIL (*Python Imaging Library*) kako bi provjerila je li slika oštećena ili neispravna. Ako slika ne zadovolji ovu provjeru, označava se kao nevažeća.

Isječak koda 2: Validacija slika

```
1         def is_valid_image(self, image_relative_path: str) -> bool:
2             try:
3                 with Image.open(full_path) as img:
4                     img.verify()
5                 if not self.detect_face(full_path):
6                     print(f"No face detected in {full_path}")
7                     self.invalid_images.append(image_relative_path)
8                     return False
9                 return True
10            except Exception as e:
11                print(f"Error with image {full_path}: {e}")
12                self.invalid_images.append(image_relative_path)
13                return False
```

Nakon provjere čitljivosti kao na slici 12, svaka slika se dodatno provjerava pomoću metode `detect_face`, koja koristi biblioteku **MediaPipe** za detekciju lica. Ako MediaPipe uspješno detektira lice, slika se smatra valjanom i može se koristiti za treniranje modela. Ako lice nije detektirano, slika se odbacuje. MediaPipe je izabran kao rješenje za detekciju lica zbog svoje preciznosti i pouzdanosti, čime je osigurano da slike koje model koristi stvarno sadrže lica.



Slika 12: Uklanjanje neispravnih slika

U metodi 3, slike se najprije učitavaju korištenjem OpenCV biblioteke i pretvaraju u RGB format (budući da OpenCV koristi BGR format po defaultu). Nakon toga, slike se obrađuju korištenjem MediaPipe biblioteke, koja detektira lica na temelju unaprijed treniranih dubokih modela. Ako se detektira lice, slika prolazi validaciju i može se koristiti za daljnje korake.

Isječak koda 3: Detekcija lica pomoću MediaPipe

```
        def detect_face(self, image_path: Path) -> bool:
            img = cv2.imread(str(image_path))
            img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            results = self.face_detection.process(img_rgb)
            if results.detections:
                return True
            return False
```

3.2.1.3. Pohrana rezultata validacije

Budući da je validacija slika dugotrajan proces, rezultati validacije (tj. popis nevaljanih slika) pohranjuju se u JSON datoteku radi bržeg učitavanja podataka u budućim iteracijama. Time se izbjegava ponovno provođenje validacije već identificiranih nevaljanih slika, što značajno smanjuje vrijeme obrade.

Isječak koda 4: Spremanje nevaljanih slika u JSON datoteku

```
1         def save_invalid_images(self, json_path="temp/invalid_images.json"):
2             if not os.path.exists("temp"):
3                 os.makedirs("temp")
4             with open(json_path, 'w') as f:
5                 json.dump(self.invalid_images, f)
```

Funkcija prikazana u kodu 4 omogućuje pohranu nevaljanih slika u datoteku kako bi se spriječilo ponovno učitavanje i obrada tih slika u budućim pokretanjima skripta. Slično tome, validirane slike također se pohranjuju kako bi se olakšao kasniji rad s tim podacima.

Isječak koda 5: Spremanje validnih slika u JSON datoteku

```
1         def save_valid_images(self, valid_images, json_path="temp/valid_images.json"
           ):
2             if not os.path.exists("temp"):
3                 os.makedirs("temp")
4             with open(json_path, 'w') as f:
5                 json.dump(valid_images, f)
```

Metode 4 i 5 omogućuju pohranu rezultata validacije slika, čime se omogućava kasniji brz pristup samo valjanim slikama, bez potrebe za ponovnim procesiranjem svih slika. Ova tehnika značajno ubrzava proces pripreme podataka, posebno kada se radi s velikim skupovima podataka kao što je IMDB-Wiki.

3.2.1.4. Filtracija slika

Jednom kada su sve slike validirane, sljedeći korak je filtracija podataka, gdje se zadržavaju samo valjane slike. Ovaj proces provodi se unutar metode `get_valid_images`, koja vraća sve validne slike te pohranjuje broj nevaljanih slika radi dokumentacije i naknadne analize.

Isječak koda 6: Filtriranje valjanih slika

```
1         def get_valid_images(self, limit=999999):
2             valid_images = []
3             for img in self.all_filenames:
4                 if len(valid_images) >= limit:
5                     break
6                 if self.is_valid_image(img):
7                     valid_images.append(img)
8             invalid_images_count = len(self.all_filenames) - len(valid_images)
9             self.save_invalid_images()
10            return valid_images, invalid_images_count
```

Metoda 6 osigurava filtriranje valjanih slika, zadržavajući one koje su prošle validaciju i detekciju lica kao na slici 13. U procesu se također pohranjuju informacije o nevaljanim slikama koje su odbačene, omogućujući analizu kvalitete podataka.

Slika 13: Validirane slike spremne za korištenje

3.2.1.5. Ekstrakcija metapodataka

Nakon validacije slika, sljedeći ključan korak bio je ekstrakcija metapodataka koji su sadržani u `.mat` datotekama za IMDB i Wiki skupove podataka. Ovi metapodaci uključuju godinu snimanja slike, datum rođenja osobe te spol, koji su ključni za treniranje modela koji predviđa dob i spol. Proces ekstrakcije metapodataka implementiran je kroz klasu `DataExtractor`. Prvo je bilo potrebno konvertirati MATLAB datume u standardni format godine, kako bi se mogla precizno izračunati starost osobe. Taj proces je implementiran statičkom metodom `_matlab_to_year`:

Isječak koda 7: Konverzija MATLAB datuma u godinu

```
1         @staticmethod
2         def _matlab_to_year(matlab_datenum: float) -> float:
3             return 1970 + (matlab_datenum - 719529) / 365.25
```

Zatim je, pomoću metode `_extract_data`, izračunata starost osobe na slici oduzimanjem godine rođenja od godine kada je slika snimljena. Podaci o dobi i spolu sortirani su u nekoliko kategorija:

- **Validni podaci**: Dob između 0 i 120 godina.
- **Negativna dob**: Podaci s negativnim vrijednostima za dob su označeni kao problematični.
- **Visoka dob**: Dob veća od 120 godina smatra se nevjerodostojnom i označava se

kao neispravan podatak.

Kombiniranjem podataka iz IMDB i Wiki skupova pomoću metode `_combine_data`, kreiran je jedinstveni skup podataka spreman za daljnju analizu. Također, problematični podaci poput negativno izračunatih godina, neobično visokih godina su dokumentirani kako bi se mogli dalje istražiti.

Isječak koda 8: Kombinacija podataka iz IMDB i Wiki skupa

```
1         def _combine_data(self) -> tuple:
2             wiki_data, wiki_invalid = self._extract_data(self.wiki_mat_data)
3             imdb_data, imdb_invalid = self._extract_data(self.imdb_mat_data)
4             return wiki_data + imdb_data, wiki_invalid + imdb_invalid
```

Ovaj proces osigurao je da su svi ključni podaci pravilno obrađeni i pripremljeni za treniranje modela, uz filtriranje neispravnih ili nelogičnih unosa.

3.2.1.6. Prikaz i vizualizacija podataka

Nakon što su podaci očišćeni i pripremljeni, važno je omogućiti vizualni pregled slika i pripadajućih metapodataka, kao i analizu distribucije podataka. U tu svrhu, klasa `DatasetAnalyzer` pruža niz metoda za jednostavan prikaz slika i vizualizaciju distribucije dobi unutar skupa podataka. Primjer prikaza specifične slike kao npr. na slici 14 zajedno s pripadajućim informacijama o dobi i spolu ostvaruje se metodom `show_image`. Ova metoda dohvaća sliku na temelju njezine relativne putanje i prikazuje je uz pripadajuće metapodatke:

Isječak koda 9: Prikaz slike s informacijama o dobi i spolu

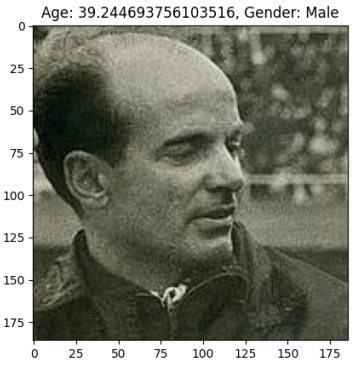
```
1         def show_image(self, index: int):
2             image_relative_path = self.df['Image_Path'][index]
3             img_path = self._get_image_path(image_relative_path)
4             img = Image.open(img_path)
5             plt.imshow(img)
6             plt.title(f"Age: {self.df['Age'][index]}, Gender: {self.df['Gender'][
               index]}")
7             plt.show()
```

Osim prikaza pojedinačnih slika, klasa omogućuje vizualizaciju distribucije godina unutar skupa podataka, što je korisno za uočavanje potencijalnih statističkih anomalija ili neravnoteže klasa. Ova vizualizacija omogućena je pomoću biblioteke `seaborn`, što olakšava uvid u distribuciju i moguće nepravilnosti u podacima:

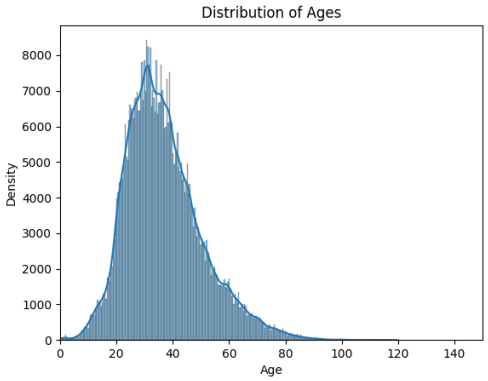
Isječak koda 10: Vizualizacija distribucije godina

```
1         def visualize_age_distribution(self):
2             sns.histplot(self.df['Age'], kde=True)
3             plt.xlabel('Age')
4             plt.ylabel('Density')
5             plt.title('Distribution of Ages')
6             plt.xlim(0, 150)
7             plt.show()
```

Vizualizacija podataka ne samo da pomaže u otkrivanju eventualnih problema s kvalitetom podataka, poput neravnomjerne raspodjele godina, već također pruža ključni uvid u balansiranost podataka za kasniji proces treniranja modela. Na primjer, prikaz distribucije dobi može ukazati na potrebu za dodatnim balansiranjem podataka između različitih dobnih skupina kako bi model bolje generalizirao na različite demografske skupine.



Slika 14: Prikaz godina i spola neke osobe



Slika 15: Distribucija godina

3.2.1.7. Odabir značajki za treniranje modela

Značajke koje su korištene za treniranje modela su dob i spol. Ove dvije demografske varijable odabrane su jer predstavljaju ključne informacije koje model treba naučiti prepoznati iz slika lica. U zadacima predikcije spola i dobi, ove značajke se često koriste zbog njihove informativnosti i jasnog značenja u kontekstu klasifikacije [24].

Proces treniranja modela zahtijeva da su značajke pravilno oblikovane kako bi model mogao učinkovito učiti. Dob je predstavljena kao kontinuirana varijabla, dok je spol binarna varijabla (0 za žene, 1 za muškarce). Ova podjela omogućava modelu da simultano uči na dvije različite vrste podataka: regresiji (predikcija dobi) i klasifikaciji (predikcija spola).

Kako bi se razumjela struktura podataka i potencijalne pristranosti, korištena je vizualizacija distribucije dobi unutar skupa podataka koja je spomenuta ranije. Ova vizualizacija je ključna jer može ukazati na neravnoteže u distribuciji dobnih skupina, što može utjecati na performanse modela. Na primjer, ako postoji značajna neravnoteža između mladih i starijih osoba, model bi mogao biti skloniji točnijoj predikciji dobi mlađih osoba, što bi moglo negativno utjecati na generalizaciju modela na starije dobne skupine.

Distribucija dobnih podataka vizualizirana je pomoću histograma, kako je prikazano u kodu 10. Ova distribucija na slici 15 omogućuje lakše prepoznavanje potencijalnih problema, poput nejednake zastupljenosti različitih dobnih skupina.

Vizualizacija distribucije omogućava uvid u to jesu li svi dobni rasponi dovoljno zastupljeni unutar skupa podataka. Ako bi se otkrila značajna neravnoteža u distribuciji dobi (npr. prevelika zastupljenost mlađih osoba), moglo bi se razmotriti uvođenje dodatnih tehnika, poput ponderiranja gubitka ili balansiranja skupa podataka kroz *oversampling* ili *undersampling*, kako bi se postigla veća uravnoteženost u treniranju modela [7].

S obzirom na prirodu podataka, posebnu pažnju treba obratiti na balansiranost podataka za spol i dob. Ako jedna kategorija značajki, poput ženskih osoba ili starijih ljudi, bude značajno manje zastupljena, model bi mogao postati pristran prema većinskoj skupini. Ovakve neravnoteže mogu uzrokovati lošije performanse modela na nerazmjerno zastupljenim skupinama, stoga se neravnoteža unutar skupa podataka redovito prati kroz vizualizaciju [16].

Na temelju ovog uvida, moglo bi se zaključiti je li potrebno dodatno balansirati skup podataka kako bi se osigurala ravnopravna zastupljenost svih kategorija i time smanjila pristranost modela.

Važno je osigurati da su značajke pravilno pripremljene i dosljedne. Tijekom procesa pripreme podataka, sve varijable (dob i spol) temeljito su provjerene kako bi se osiguralo da ne sadrže nedostajuće ili pogrešne vrijednosti. Ovaj proces osigurava da model trenira na točnim podacima, što pridonosi njegovoj točnosti i robusnosti. Korišteni su relevantni podaci, pri čemu su uklonjene nepravilnosti poput netočnih metapodataka o datumu rođenja ili spolu, čime se postigla veća kvaliteta podataka [25].

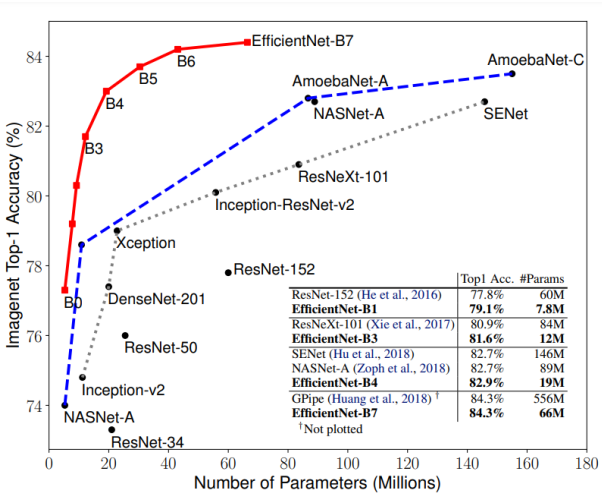
4. Primjena modela na prepoznavanje dobi i spola

U ovom poglavlju detaljno je opisan razvoj i primjena modela za prepoznavanje dobi i spola s ciljem da se postignu visoke performanse u zadatku klasifikacije spola te regresije dobi. Odluke prilikom odabira modela, postavljanja eksperimenta i treniranja utemeljene su na industrijskim standardima, s naglaskom na robusnost i stabilnost modela u stvarnim primjenama računalnog vida.

4.1. Odabir modela

Odabir modela ključan je korak u svakom zadatku dubokog učenja jer izravno utječe na performanse i generalizacijsku sposobnost modela. Za zadatak prepoznavanja spola i dobi, korištena je arhitektura *EfficientNet-B4*, koja je jedna od najsuvremenijih konvolucijskih neuronskih mreža predtreniranih na skupu podataka *ImageNet*. Ovaj model sa slike 16 odabran je zbog nekoliko ključnih razloga:

- **Učinkovitost**: *EfficientNet-B4* koristi optimiziranu arhitekturu koja balansira točnost i brzinu, čineći je idealnom za primjene gdje su resursi ograničeni.
- **Predtreniranost na ImageNetu**: ImageNet je jedan od najraširenijih i najsloženijih skupova podataka u računalnom vidu. Predtreniranje na ovom skupu omogućuje modelu da već ima ugrađeno znanje o osnovnim značajkama slika, što ubrzava proces prilagodbe modela za nove zadatke, poput prepoznavanja spola i dobi.
- **Složena arhitektura**: *EfficientNet* koristi višedimenzionalnu optimizaciju koja skalira dubinu, širinu i rezoluciju modela na uravnotežen način, što doprinosi visokoj točnosti uz relativno mali broj parametara.



Slika 16: Veličina modela u odnosu na ImageNet točnost; preuzeto iz [26]

Modificirana verzija *EfficientNet-B4* prilagođena je za zadatak istovremene predikcije spola (kao klasifikacijskog problema) i dobi (kao regresijskog problema). Ova prilagodba podrazumijeva izmjenu završnog sloja modela kako bi se podržala dva različita izlaza – jedan za regresiju dobi, a drugi za klasifikaciju spola. Ovaj pristup omogućuje paralelno učenje dvaju zadataka, što je česta praksa u složenim primjenama računalnog vida. Kod za implementaciju ovakvog modela prikazan je u isječku 11.

Isječak koda 11: Modifikacija EfficientNet-B4 za predikciju dobi i spola

```
class AgeGenderModel(torch.nn.Module):
    def __init__(self):
        super(AgeGenderModel, self).__init__()
        self.efficientnet = models.efficientnet_b4(weights=models.
         EfficientNet_B4_Weights.IMAGENET1K_V1)
        self.efficientnet.classifier[1] = torch.nn.Linear(self.efficientnet.
         classifier[1].in_features, 1024)
        self.dropout = torch.nn.Dropout(p=0.5)
        self.fc_age = torch.nn.Linear(1024, 1)
        self.fc_gender = torch.nn.Linear(1024, 2)

    def forward(self, x):
        features = self.efficientnet(x)
        features = self.dropout(features)
        age = self.fc_age(features)
        gender = self.fc_gender(features)
        return age, gender
```

Ovdje je vidljivo da završni slojevi *EfficientNet-B4* uključuju dvije linije za predikciju: jednu koja koristi linearni sloj za regresiju dobi i drugu koja koristi linearni sloj za klasifikaciju spola. Slojevi se koriste nakon dodavanja *dropout* sloja, čime se model čini otpornijim na pretreniranost, što je ključan čimbenik u modelima koji se treniraju na malim ili neuravnoteženim skupovima podataka.

Arhitektura *EfficientNet* predstavlja značajan napredak u dizajnu CNN modela zbog svoje sposobnosti da balansira dubinu, širinu i rezoluciju modela na optimalan način. Tradicionalni CNN modeli često povećavaju performanse modela samo povećanjem dubine, što može dovesti do prekomjernog broja parametara bez proporcionalnog povećanja točnosti. *EfficientNet* je uveo metodu *Compound Scaling*, koja simultano skalira sve tri dimenzije modela – dubinu (*depth*), širinu (*width*) i rezoluciju (*resolution*) [26].

Matematički, broj operacija modela (FLOPS) proporcionalan je produktu dubine, kvadrata širine i kvadrata rezolucije, prema sljedećoj formuli:

$$\text{FLOPS} \propto \text{depth} \cdot \text{width}^2 \cdot \text{resolution}^2$$

Taj pristup [26] omogućuje da se kapacitet modela proširuje proporcionalno sa svakom dimenzijom, čime se postiže uravnotežen rast performansi bez nepotrebnog povećanja broja parametara. U sklopu tog pristupa, *EfficientNet-B4* pruža optimalan omjer između točnosti i učinkovitosti, čime se omogućava primjena u zadacima s ograničenim resursima.

Model *EfficientNet* također koristi *MBConv* slojeve, koji su optimizirani za mobilne uređaje i omogućuju visoku učinkovitost u smanjenju broja parametara. Svaki *MBConv* sloj uključuje uske i ekspanzijske slojeve koji omogućuju efikasnu kompresiju i ekspanziju značajki. Također, aktivacijska funkcija *Swish* [27], koja se koristi unutar modela, zamjenjuje tradicionalne *ReLU* funkcije. *Swish* funkcija definira se kao:

$$\text{Swish}(x) = x \cdot \sigma(x)$$

gdje je $\sigma(x)$ standardna sigmoidna funkcija. Ova funkcija pomaže modelu da bolje uči nelinearne odnose, što pridonosi većoj točnosti u zadacima klasifikacije i regresije. Za zadatak predikcije spola i dobi, model *EfficientNet-B4* prilagođen je tako da podržava dva različita izlaza – jedan za regresiju dobi i drugi za klasifikaciju spola. Kao što je prikazano u isječku koda 11, zadnji sloj modela zamijenjen je s dva linearna sloja: jedan za predikciju kontinuirane varijable (dob), a drugi za binarnu klasifikaciju spola. U ovoj implementaciji, sloj *dropout* dodan je kako bi se smanjila mogućnost pretreniranosti modela. *Dropout* isključuje određeni postotak neurona tijekom svakog prolaza treniranja, čime se model prisiljava da nauči robusnije značajke, a ne da se oslanja na specifične neuronske jedinice. *Dropout* sloj u ovom slučaju koristi se s postotkom isključivanja od 50%, čime se postiže balans između složenosti modela i njegove sposobnosti generalizacije.

Matematički, regresija dobi definirana je sljedećom formulom [1, str. 170–171]:

$$\hat{y}_{age} = W_{age} \cdot f(x) + b_{age}$$

gdje je $f(x)$ niz značajki dobivenih prolaskom slike kroz model *EfficientNet*, dok su W_{age} i b_{age} težine i pristranost linearne regresije. Ovdje $\hat{y}_{age}$ predstavlja predikciju dobi. Za predikciju spola koristi se softmax funkcija [1, str. 81]:

$$\hat{y}_{gender} = \text{softmax}(W_{gender} \cdot f(x) + b_{gender})$$

gdje je softmax definirana kao:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

gdje je z_i logit za i-tu klasu, a K broj klasa (u ovom slučaju dvije klase: muško i žensko). Odabir modela *EfficientNet-B4* za zadatak predikcije dobi i spola pokazao se kao optimalan izbor s obzirom na učinkovitost, točnost i skalabilnost. Korištenjem metode *Compound Scaling*, model uspijeva postići visoke performanse s relativno malim brojem parametara. Predtreniranje na *ImageNetu* osigurava da model već ima osnovno znanje o značajkama slika, dok prilagodba modela s dodatnim slojevima za predikciju dobi i spola omogućuje uspješnu primjenu u složenim zadacima računalnog vida.

4.2. Treniranje modela

Treniranje modela predstavlja jedan od najvažnijih koraka u razvoju svakog modela strojnog učenja. U ovom slučaju, model je dizajniran za istovremenu predikciju dobi (regresijski zadatak) i spola (klasifikacijski zadatak). Implementacija treniranja uključuje specifične korake koji su važni za postizanje visoke točnosti i generalizacijske sposobnosti modela. Ovi koraci uključuju dizajniranje odgovarajućih transformacija, definiciju skupa podataka, odabir odgovarajućih *loss* funkcija i postavljanje odgovarajućih hiperparametara.

Glavni izazov pri treniranju ovog modela bio je istovremeno rješavanje dva različita zadatka. S jedne strane, model mora predvidjeti dob, što je kontinuirana varijabla, dok s druge strane model mora predvidjeti spol, što je klasifikacijski zadatak s dvije klase (muško i žensko). Zbog toga je bilo nužno implementirati različite strategije za rješavanje svakog od tih zadataka te ih kombinirati unutar jedinstvenog modela.

Model je treniran na 20 epoha s korištenjem *Adam* optimizatora, a evaluacija se provodila nakon svake epohe kako bi se mjerila točnost i preciznost modela na neviđenim podacima. Parametri poput stope učenja i raspada težina (*weight decay*) bili su ključni za sprječavanje pretreniranosti i postizanje optimalnih rezultata na skupu podataka.

4.2.1. Postavljanje eksperimenta

Postavljanje eksperimenta počinje od pripreme podataka. Prije treniranja modela, podaci moraju biti unaprijed obrađeni i pripremljeni na način koji omogućuje modelu da iz njih izvuče korisne informacije. Nakon učitavanja ispravnih slika odnosno podataka, podaci su podijeljeni u skup za treniranje i skup za testiranje u omjeru 80:20. Ova podjela osigurava da model ima dovoljno primjera za učenje, ali i za evaluaciju performansi na neviđenim podacima. Korišten je `train_test_split` iz biblioteke `sklearn` kako bi se postigla ova podjela:

Isječak koda 12: Podjela podataka na skup za treniranje i testiranje

```
1 train_data, test_data = train_test_split(data, test_size=0.2, random_state
   =42)
```

Također, postavljanje eksperimenta uključivalo je definiranje transformacija koje su korištene kako bi se model učinio robusnijim. Transformacije su ključne jer simuliraju razne scenarije koji se mogu pojaviti u stvarnim uvjetima snimanja slika. U ovom eksperimentu korištene su sljedeće transformacije:

- **Nasumične promjene rezolucije** (*RandomResizedCrop*): omogućuje modelu da postane otporan na slike različitih veličina i omjera.
- **Horizontalno zrcaljenje** (*RandomHorizontalFlip*): omogućuje modelu da nauči prepoznati lice neovisno o smjeru u kojem je okrenuto.
- **Promjene svjetline, kontrasta i zasićenja** (*ColorJitter*): ove transformacije pomažu modelu da postane otporniji na različite uvjete osvjetljenja i kvalitete slike.

- **Normalizacija** (*Normalize*): normalizacija omogućava konzistentnost u ulaznim podacima, smanjujući varijabilnost uzrokovanu različitim rasponima piksela.

Kod za definiranje ovih transformacija prikazan je u isječku 13.

Isječak koda 13: Transformacije podataka za treniranje modela

```
train_transform = transforms.Compose([
    transforms.RandomResizedCrop(224),
    transforms.RandomHorizontalFlip(),
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue
     =0.2),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

test_transform = transforms.Compose([
    transforms.Resize(224),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

Korištenjem ovih transformacija, model je treniran na različitim varijacijama ulaznih slika, čime se poboljšala njegova sposobnost generalizacije i otpornosti na nepravilnosti koje mogu nastati u stvarnim slikama. Korištene su transformacije specifične za treniranje (*train_transform*), dok su za skup za testiranje korištene konzistentne transformacije (*test_transform*) koje osiguravaju da model evaluira slike pod uvjetima koji su u skladu sa stvarnim slikama.

Nakon definiranja transformacija, kreirani su `DataLoader`-i, koji su odgovorni za učinkovito učitavanje i obradu podataka. `DataLoader`-i omogućuju brzi prolazak kroz velike količine podataka tijekom treniranja i evaluacije. U slučaju ovog modela, slike i njihovi pripadajući metapodaci (dob i spol) obrađuju se u serijama od četiri slike za treniranje, te u serijama od dvije slike za testiranje:

Isječak koda 14: Definiranje `DataLoader`-a

```
train_loader = DataLoader(train_dataset, batch_size=4, shuffle=True,
 num_workers=8)
test_loader = DataLoader(test_dataset, batch_size=2, shuffle=False,
 num_workers=8)
```

Shuffle parametar je postavljen na *True* tijekom treniranja kako bi se osiguralo da se podaci nasumično odabiru za svaku epohu, čime se smanjuje mogućnost pretreniranosti.

4.2.2. Parametri treniranja

Parametri treniranja ključni su za postizanje optimalnih performansi modela. Nakon definiranja skupa podataka i transformacija, fokus je bio na definiranju ključnih hiperparametara kao što su broj epoha, stopa učenja, optimizator i *weight decay*. Svaki od tih parametara pažljivo je odabran kako bi se osiguralo pravilno treniranje modela.

Broj epoha definira koliko puta model prolazi kroz cijeli skup podataka za treniranje. U ovom eksperimentu model je treniran na 20 epoha, vremenski negdje oko 21 sat (s obzirom na specifikacije računala na kojem se trenirao model). Ovaj broj epoha odabran je na temelju industrijskih praksi za treniranje velikih modela poput *EfficientNet-a*, s obzirom na to da ovaj broj epoha obično osigurava dovoljan broj iteracija za učenje bez značajnog rizika od pretreniranosti. Pretreniranost se pojavljuje kada model nauči previše specifičnih značajki iz skupa podataka za treniranje, zbog čega mu opada sposobnost generalizacije na neviđene podatke.

Stopa učenja jedan je od najvažnijih hiperparametara u treniranju modela. Ona određuje koliko brzo model prilagođava svoje težine tijekom svakog koraka optimizacije. U ovom eksperimentu, stopa učenja postavljena je na 0.001, što je vrijednost koja se često koristi u praksi za treniranje modela poput *EfficientNet-a*. Korištena je konstantna stopa učenja kroz cijelo treniranje, što je omogućilo stabilno učenje bez prevelikih oscilacija u vrijednostima gubitka.

Za optimizaciju težina modela korišten je *Adam* optimizator, što je optimizator baziran na gradijentnom spuštanju. *Adam* optimizator široko se koristi u treniranju dubokih modela zbog svoje sposobnosti brzog konvergiranja i automatiziranog prilagođavanja stope učenja. Ovaj optimizator koristi informacije o prvom i drugom momentu gradijenata kako bi prilagodio stope učenja za svaki parametar modela, čime se omogućuje brže učenje i stabilnost.

Kod koji prikazuje definiranje optimizatora i parametara treniranja nalazi se u isječku 15.

Isječak koda 15: Treniranje modela uz definirane parametre

```
class Trainer:
    def __init__(self, model, train_loader, test_loader, device, num_epochs
     =20, weight_decay=1e-4):
        self.model = model
        self.train_loader = train_loader
        self.test_loader = test_loader
        self.device = device
        self.num_epochs = num_epochs
        self.criterion_age = torch.nn.MSELoss()
        self.criterion_gender = torch.nn.CrossEntropyLoss()
        self.optimizer = torch.optim.Adam(self.model.parameters(), lr=0.001,
          weight_decay=weight_decay)

    def train(self):
        self.model.to(self.device)
        for epoch in range(self.num_epochs):
            print(f"Starting epoch {epoch+1}/{self.num_epochs}")
            self.model.train()
            running_loss = 0.0
            for i, (images, ages, genders) in enumerate(self.train_loader):
                images, ages, genders = images.to(self.device), ages.to(self
                 .device), genders.to(self.device)
                self.optimizer.zero_grad()
                outputs_age, outputs_gender = self.model(images)
                loss_age = self.criterion_age(outputs_age.squeeze(), ages)
                loss_gender = self.criterion_gender(outputs_gender, genders)
                loss = loss_age + loss_gender
                loss.backward()
                self.optimizer.step()
                running_loss += loss.item()
            avg_train_loss = running_loss / len(self.train_loader)
            avg_test_loss = self.evaluate()
            print(f"Epoch [{epoch+1}/{self.num_epochs}], Train Loss: {
             avg_train_loss:.4f}, Test Loss: {avg_test_loss:.4f}")
        torch.save(self.model.state_dict(), 'results/age_gender.pth')
```

Tijekom svakog prolaska (epohe), model ažurira svoje težine na temelju pogrešaka koje izračunava pomoću dvije funkcije gubitka (*MSE* za regresiju i *CrossEntropy* za klasifikaciju). Na taj način model uči značajke koje su relevantne za predikciju dobi i spola.

Raspad težina (*weight decay*) odabran je kako bi se smanjio problem pretreniranosti modela. U ovom eksperimentu, *weight decay* postavljen je na $1e^{-4}$, što znači da model primjenjuje regularizaciju na težine tijekom treniranja kako bi se izbjegla prevelika prilagodba težina podacima za treniranje. Ova tehnika pomaže modelu da bolje generalizira na neviđene podatke, smanjujući rizik od prevelike osjetljivosti na specifične primjere iz skupa podataka za treniranje.

Funkcije gubitka ključni su element u treniranju bilo kojeg modela jer definiraju kako se mjeri pogreška modela. U ovom eksperimentu korištene su dvije različite funkcije gubitka:

- **Srednja kvadratna pogreška (*MSE*)** za predikciju dobi: *MSE* se koristi za regresijske zadatke gdje se mjeri razlika između predviđene i stvarne vrijednosti.
- **Unakrsna entropija (*CrossEntropy*)** za predikciju spola: *CrossEntropy* se koristi za klasifikacijske zadatke s više klasa, a mjeri udaljenost između predviđene distribucije vjerojatnosti i stvarne klase.

Isječak koda 16: Funkcije gubitka za regresiju i klasifikaciju

```
1         self.criterion_age = torch.nn.MSELoss()
2         self.criterion_gender = torch.nn.CrossEntropyLoss()
```

Ove funkcije gubitka omogućuju modelu da paralelno uči na oba zadatka, istovremeno optimizirajući svoju sposobnost za predikciju kontinuirane vrijednosti (dob) i klasifikacije (spol). Treniranje modela predstavljalo je izazovan i iterativan proces koji je uključivao pažljivo postavljanje hiperparametara, optimizaciju modela i prilagodbu na specifične zadatke. Korištenjem *EfficientNet-B4* modela, postignute su dobre performanse na zadacima prepoznavanja dobi i spola. Transformacije podataka i pažljivo podešeni hiperparametri osigurali su robusnost i otpornost modela na varijacije u podacima, dok su korištene funkcije gubitka omogućile uspješno paralelno učenje oba zadatka.

4.3. Evaluacija modela

Nakon treniranja modela, ključni korak u razvoju svakog sustava strojnog učenja je temeljita evaluacija performansi modela. Evaluacija pruža uvid u to koliko dobro model generalizira na neviđene podatke, je li otporan na razne vrste smetnji, te kolika je njegova robusnost u stvarnim uvjetima primjene. Ovaj dio rada fokusira se na tri ključne metrike: performanse modela, stabilnost i robusnost.

4.3.1. Performanse

Evaluacija performansi modela prvi je i najvažniji aspekt evaluacije. Performanse modela mjere se u smislu njegove sposobnosti da pravilno predvidi dob (regresija) i spol (klasifikacija) na skupu za testiranje, koji model nikada nije vidio tijekom treniranja.

Kod za evaluaciju modela implementiran je kroz funkciju `evaluate`, koja koristi skup podataka za testiranje kako bi izračunala ukupni gubitak i prikazala predikcije u usporedbi s stvarnim vrijednostima. Performanse modela izražavaju se u terminima gubitka (*loss*), koji mjeri razliku između stvarnih vrijednosti i predikcija modela. Za dob se koristi srednja kvadratna pogreška (*MSE*), dok se za spol koristi unakrsna entropija (*CrossEntropy*). U nastavku je prikazan isječak koda 17 za evaluaciju modela:

Isječak koda 17: Evaluacija performansi modela

```
1         def evaluate(self):
2             self.model.eval()
3             test_loss = 0.0
4             with torch.no_grad():
5                 for images, ages, genders in self.test_loader:
6                     if images is None:
7                         continue
8                     images, ages, genders = images.to(self.device), ages.to(self.
                       device), genders.to(self.device)
9                     outputs_age, outputs_gender = self.model(images)
10
11                    loss_age = self.criterion_age(outputs_age.squeeze(), ages)
12                    loss_gender = self.criterion_gender(outputs_gender, genders)
13                    loss = loss_age + loss_gender
14
15                    test_loss += loss.item()
16
17                    print(f"Predicted Age: {outputs_age.squeeze().cpu().numpy()}")
18                    print(f"Predicted Gender: {torch.argmax(outputs_gender, dim=1).
                       cpu().numpy()}")
19                    print(f"Actual Age: {ages.cpu().numpy()}")
20                    print(f"Actual Gender: {genders.cpu().numpy()}")
21
22            return test_loss / len(self.test_loader)
```

U ovoj funkciji `evaluate`, model prelazi u način rada za evaluaciju pomoću metode `model.eval()`, koja isključuje određene slojeve, poput *dropout*-a, koji se koriste tijekom treniranja radi regularizacije. Na ovaj način model izračunava predikcije bez dodatnih varijacija. Svaka iteracija kroz skup za testiranje uključuje izračun gubitka za dob i spol te usporedbu predikcija s stvarnim vrijednostima.

Izračun ukupnog gubitka pomaže identificirati koliko se model udaljava od stvarnih vrijednosti, dok se prikaz predikcija koristi za kvalitativnu analizu rezultata. U ovom slučaju, dob se predviđa kao kontinuirana varijabla (regresija), dok se spol predviđa kao diskretna varijabla (klasifikacija) između dvije klase: muško i žensko. Točnost klasifikacije spola može se lako izračunati usporedbom predikcija s stvarnim vrijednostima, dok za dob koristimo metriku pogreške, kao što je srednja kvadratna pogreška (*MSE*), kako bismo mjerili koliko su predikcije udaljene od stvarnih godina.

Nakon što model završi evaluaciju na cijelom skupu za testiranje, ukupni gubitak se vraća kao mjera performansi. Ovaj testni gubitak služi kao ključna mjera za procjenu koliko je model uspješan u generalizaciji na neviđene podatke.

4.3.2. Stabilnost

Stabilnost modela odnosi se na sposobnost modela da zadrži svoje performanse kada se suoči s malim promjenama u ulaznim podacima. Drugim riječima, model bi trebao davati konzistentne rezultate čak i kada su podaci blago izmijenjeni, primjerice kada se doda šum, kada se slike rotiraju ili kada se promijene uvjeti osvjetljenja. Ovaj aspekt stabilnosti posebno je važan u stvarnim primjenama, gdje se slike rijetko pojavljuju u savršenim uvjetima.

Kako bi se procijenila stabilnost modela, razvijena je funkcija `test_stability`. Ova funkcija dodaje razne vrste smetnji originalnim slikama iz skupa za testiranje, a zatim model predviđa dob i spol na temelju izmijenjenih slika. Funkcija bilježi originalne predikcije i uspoređuje ih s predikcijama dobivenim iz izmijenjenih slika kako bi se utvrdilo koliko su promjene u predikcijama značajne. U nastavku su prikazani isječci 18, 19 za procjenu stabilnosti modela:

Isječak koda 18: Testiranje stabilnosti modela

```
1         def test_stability(self):
2             self.model.eval()
3             with torch.no_grad():
4                 for i, (images, ages, genders) in enumerate(self.test_loader):
5                     if images is None:
6                         continue
7                     images, ages, genders = images.to(self.device), ages.to(self.
                       device), genders.to(self.device)
8                     original_age_predictions, original_gender_predictions = self.
                       model(images)
9
10                    # Add Gaussian noise
11                    noisy_images = images + torch.randn_like(images) * 0.1
12                    noisy_age_predictions, noisy_gender_predictions = self.model(
                       noisy_images)
```

Isječak koda 19: Testiranje stabilnosti modela (*cont.*)

```
                # Rotate images
                rotated_images = torch.rot90(images, k=1, dims=[2, 3])
                rotated_age_predictions, rotated_gender_predictions = self.model
                 (rotated_images)

                # Adjust brightness
                brightness_images = torch.clamp(images * 1.5, 0, 1)
                brightness_age_predictions, brightness_gender_predictions = self
                 .model(brightness_images)

                # Adjust contrast
                contrast_images = torch.clamp((images - 0.5) * 1.5 + 0.5, 0, 1)
                contrast_age_predictions, contrast_gender_predictions = self.
                 model(contrast_images)

                for j in range(len(images)):
                    self.stability_log.append([
                        i+1, j,
                        original_age_predictions[j].cpu().numpy(),
                         original_gender_predictions[j].cpu().numpy(),
                        noisy_age_predictions[j].cpu().numpy(),
                         noisy_gender_predictions[j].cpu().numpy(),
                        rotated_age_predictions[j].cpu().numpy(),
                         rotated_gender_predictions[j].cpu().numpy(),
                        brightness_age_predictions[j].cpu().numpy(),
                         brightness_gender_predictions[j].cpu().numpy(),
                        contrast_age_predictions[j].cpu().numpy(),
                         contrast_gender_predictions[j].cpu().numpy()
                    ])
                    self.save_logs()
                torch.cuda.empty_cache()
```

Ovaj kod omogućava testiranje stabilnosti modela prema sljedećim promjenama:

- **Šum** (*Gaussian noise*): Dodaje se nasumičan šum na slike kako bi se testiralo kako model reagira na manje vizualne smetnje.
- **Rotacija**: Slike se rotiraju za 90 stupnjeva kako bi se provjerilo kako model reagira na promjene orijentacije slike.
- **Svjetlina i kontrast**: Slike se mijenjaju promjenom svjetline i kontrasta kako bi se simulirali uvjeti snimanja u različitim osvjetljenjima.

Svaka od ovih promjena može imati različit utjecaj na performanse modela, a bilježenjem predikcija prije i nakon tih promjena moguće je dobiti uvid u to koliko su predikcije stabilne. U slučaju da model pokaže značajne promjene u predikcijama na samo malim izmjenama ulaznih podataka, to bi značilo da model nije stabilan i da je osjetljiv na male smetnje, što bi moglo biti problematično u stvarnim primjenama.

4.3.3. Robusnost

Osim stabilnosti, model mora biti i robusan, što znači da treba zadržati svoje performanse i kada se suoči s ciljanom neprijateljskom interferencijom, poput *adversarial* primjera. *Adversarial* primjeri su slike koje su posebno dizajnirane kako bi "zbunile" model, tj. da ga navedu da napravi pogrešne predikcije. U stvarnim primjenama, robusnost modela postaje sve važnija, osobito u sigurnosno osjetljivim područjima.

Za procjenu robusnosti modela, korištena je metoda napada *PGD* (*Projected Gradient Descent*), koja generira *adversarial* primjere. Ovi primjeri se zatim koriste za testiranje robusnosti modela prema namjernim interferencijama. Cilj ove evaluacije je vidjeti koliko je model otporan na pokušaje manipulacije predikcijama kroz namjerne, specifične promjene u slikama. U nastavku je prikazan isječak koda 20 za testiranje robusnosti modela:

Isječak koda 20: Testiranje robusnosti modela pomoću PGD napada

```
1         def test_robustness(self):
2             gender_model = GenderPredictionModel(self.model)
3             gender_model.to(self.device)
4             attack = torchattacks.PGD(gender_model, eps=0.1, alpha=0.01, steps=40)
5
6             for i, (images, ages, genders) in enumerate(self.test_loader):
7                 if images is None:
8                     continue
9                 images, ages, genders = images.to(self.device), ages.to(self.device)
                   , genders.to(self.device)
10                original_age_predictions, original_gender_predictions = self.model(
                   images)
11
12                adversarial_images = attack(images, genders)
13                adversarial_age_predictions, adversarial_gender_predictions = self.
                   model(adversarial_images)
14
15                for j in range(len(images)):
16                    self.robustness_log.append([
17                        i+1, j,
18                        original_age_predictions[j].detach().cpu().numpy(),
                           original_gender_predictions[j].detach().cpu().numpy(),
19                        adversarial_age_predictions[j].detach().cpu().numpy(),
                           adversarial_gender_predictions[j].detach().cpu().numpy()
20                    ])
21                    self.save_logs()
22                torch.cuda.empty_cache()
```

Kod za testiranje robusnosti koristi *PGD* napad, jedan od najpoznatijih napada u domenu *adversarial* primjera. *PGD* napad radi iterativno i koristi gradijente kako bi unio male promjene u ulazne slike s ciljem da promjene predikciju modela. Ovdje je korišten napad s parametrima:

- **Epsilon (*eps*)**: Veličina promjene na slici (u ovom slučaju 0.1).

- **Alpha**: Količina promjene u svakoj iteraciji (0.01).
- **Koraci (*steps*)**: Broj iteracija (40 koraka).

Nakon generiranja *adversarial* primjera, model testira predikcije na tim izmijenjenim slikama i bilježi rezultate. Robusnost modela može se procijeniti usporedbom originalnih predikcija s predikcijama na *adversarial* primjere. Ako model daje značajno različite predikcije na *adversarial* primjere, to bi značilo da je model podložan manipulacijama i nije dovoljno robustan.

5. Rezultati i analiza

U ovom poglavlju analizira se preciznost modela u predviđanju dobi i spola, procjenjuju performanse na testnom skupu, te se ispituje stabilnost i robusnost modela pod različitim uvjetima. Kroz podsekcije analiziraju se distribucije podataka, točnost predikcija, te kako perturbacije poput šuma, rotacije i promjena osvjetljenja utječu na predikcije. Također, ispituje se otpornost modela na *adversarial* napade i mjeri koliko predikcije ostaju točne pod takvim uvjetima. Na kraju, procjenjuje se modelova sposobnost konvergencije tijekom treniranja i uspješnost na testnom skupu.

5.1. Statistička analiza numeričkih predikcija i međukoraka

Statistička analiza numeričkih predikcija i međukoraka prilikom treniranja provedena je u programskom jeziku **Julia**. Analize u programskom jeziku Julia obuhvaćaju učitavanje i obradu podataka, izvođenje statističkih testova, te vizualizaciju rezultata. Korištenje paketa `DataFrames`, `HypothesisTests`, `GLM` i `Plots` omogućava brzu i efikasnu analizu modela. Primjer koda za analizu predikcija dobi i spola uključuje funkcije za izračun osnovne statistike, korelacije, regresijske analize te metrike kao što su *RMSE* i *MAE*.

Isječak koda 21: Primjer analize u programskom jeziku Julia

```
using CSV, DataFrames, Statistics, Plots, HypothesisTests, GLM

# Učitavanje podataka
function load_data(file_path::String)::DataFrame
    data = CSV.read(file_path, DataFrame)
    rename!(data, "Predicted Age" => :Predicted_Age, "Actual Age" => :Actual_Age
     )
    return data
end

# Izračun osnovne statistike
function get_basic_statistics(data::DataFrame)
    return describe(data)
end

# Korelacija između stvarne i predviđene dobi
function calculate_correlation(data::DataFrame)::Float64
    return cor(data.Predicted_Age, data.Actual_Age)
end
```

Isječak koda 22: Primjer analize u programskom jeziku Julia (*cont.*)

```
# Linearna regresija
function perform_linear_regression(data::DataFrame)::StatsModels.
 TableRegressionModel
    return lm(@formula(Actual_Age ~ Predicted_Age), data)
end

# RMSE i MAE
function calculate_errors(data::DataFrame)
    residuals = data.Predicted_Age - data.Actual_Age
    rmse = sqrt(mean(residuals .^ 2))
    mae = mean(abs.(residuals))
    return rmse, mae
end

# Vizualizacija rezultata
function plot_scatter_with_regression(data::DataFrame, model::StatsModels.
 TableRegressionModel)
    plt = scatter(data.Predicted_Age, data.Actual_Age, title="Predviđena vs
     Stvarna dob")
    plot!(plt, data.Predicted_Age, fitted(model), label="Regresijska linija")
    savefig(plt, "scatter_with_regression.png")
end
```

Ovaj primjer koda pokazuje kako se obavlja osnovna statistička analiza podataka, izračunavaju korelacije i greške te kako se vizualiziraju rezultati koristeći funkcije za grafičko prikazivanje. Analize pružaju ključne informacije o performansama modela i mogućim poboljšanjima.

5.1.1. Analiza podataka dobi i spola

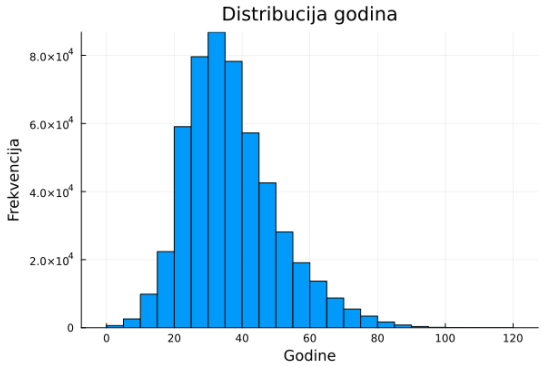
Analiza podataka o dobi i spolu ključna je za razumijevanje demografskih karakteristika skupa podataka. Cilj je identificirati distribucije dobi i spola, te analizirati njihov međusobni odnos kroz statističke testove, čime se procjenjuju potencijalne razlike između spolova. Time se olakšava prepoznavanje potencijalnih pristranosti, što je važno za pravilno treniranje modela.

Prva faza obuhvaća osnovnu statističku analizu podataka, pri čemu su podaci o dobi kvantitativni, a podaci o spolu kategorijski. Tablica 1 prikazuje osnovne mjere za dob i spol u skupu podataka. Prosječna dob iznosi 36.91 godina, s minimalnom dobi od 0 i maksimalnom od 119.62. Nema nedostajućih vrijednosti, što osigurava kvalitetu podataka.

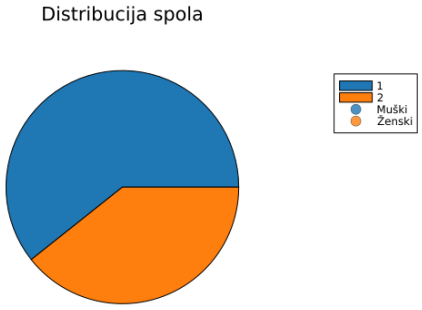
Varijabla	Prosjek	Minimum	Medijan	Maksimum	Nedostajuće vrijednosti
Dob	36.91	0.00	34.98	119.62	0
Spol	-	Ženski	-	Muški	0

Tablica 1: Osnovna statistika za dob i spol

Distribucije dobi i spola prikazane su na slikama 17 i 18. Histogram dobi pokazuje ravnomjernu distribuciju, uz vrhunce u nižim dobnim skupinama. Primjetni su neki ekstremi u starijim godinama, što zahtijeva pažljivije razmatranje prilikom treniranja modela.



Slika 17: Distribucija dobi



Slika 18: Distribucija spola

Distribucija spola ukazuje na blagu dominaciju muških uzoraka (60.68%) u odnosu na ženske (39.32%). Ravnoteža među spolovima ključna je kako bi se spriječila pristranost modela prema većinski zastupljenoj skupini.

Za procjenu razlike u godinama između spolova, korišten je T-test s jednakom varijancom. Rezultati T-testa ukazuju na značajnu razliku između muških i ženskih dobnih skupina ($p < 1 \times 10^{-99}$), s prosječnom razlikom od 6.08 godina. To sugerira da su muškarci u prosjeku stariji od žena u ovom skupu podataka.

```
T-test rezultat:
Two sample t-test (equal variance)
---------------------------------
Population details:
    parameter of interest:   Mean difference
    value under h_0:         0
    point estimate:          6.08166
    95% confidence interval: (6.008, 6.156)

Test summary:
    outcome with 95% confidence: reject h_0
    two-sided p-value:           <1e-99

Details:
    number of observations:   [316044, 204795]
    t-statistic:              160.76
    degrees of freedom:       520837
    empirical standard error: 0.03783
```

Rezultati analize, dakle, ukazuju na demografske razlike koje mogu utjecati na treniranje modela. Model bi mogao razviti pristranost prema spolu ili određenim dobnim skupinama, pa je potrebno poduzeti korake poput balansiranja podataka ili primjene metoda za smanjenje pristranosti. Ispravna analiza podataka ključna je za kvalitetnu izgradnju modela.

5.1.2. Evaluacija predikcija

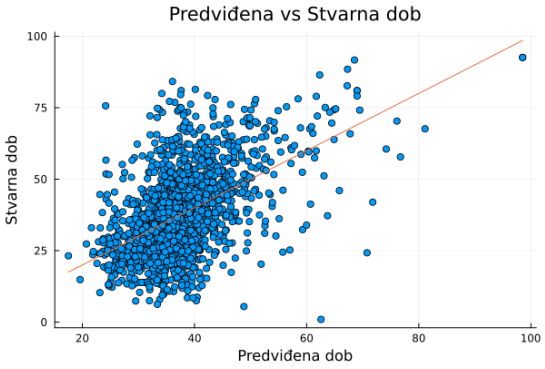
Evaluacija predikcija ključna je za procjenu učinkovitosti modela u predviđanju stvarnih vrijednosti. U ovom slučaju, analizirala se točnost modela u predviđanju dobi i spola. Cilj je bio usporediti predviđene vrijednosti s njihovim stvarnim vrijednostima, kvantificirati pogreške te evaluirati uspješnost klasifikacije spola kroz metrike kao što su točnost, preciznost i F1-mjera.

Tablica 2 prikazuje osnovne statističke mjere za predviđenu i stvarnu dob te predviđeni i stvarni spol. Vidimo da je prosječna predviđena i stvarna dob vrlo blizu, 37.49 godina, što ukazuje na relativno dobru preciznost modela. Prosječna vrijednost predviđenog spola je 0.80, dok je stvarna 0.60, što također ukazuje na dobru klasifikaciju.

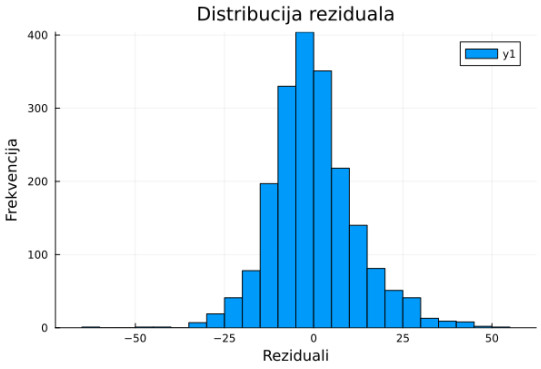
Varijabla	Prosjek	Minimum	Medijan	Maksimum	Nedostajuće vrijednosti
Predviđena dob	37.49	17.51	36.68	98.52	0
Stvarna dob	37.49	0.95	35.57	92.57	0
Predviđeni spol	0.80	0	1.00	1.00	0
Stvarni spol	0.60	0	1.00	1.00	0

Tablica 2: Osnovna statistika za predviđenu i stvarnu dob te spol

Korelacija između predviđene i stvarne dobi iznosi 0.618, što ukazuje na solidnu povezanost između ovih dviju varijabli, no ipak ukazuje na određenu količinu pogrešaka u predikcijama. Graf raspršenosti s regresijskom linijom na slici 19 dodatno prikazuje odnos između predviđene i stvarne dobi, gdje se može vidjeti blaga disperzija oko regresijske linije, što ukazuje na postojanje određenih odstupanja.



Slika 19: Graf raspršenosti



Slika 20: Histogram reziduala

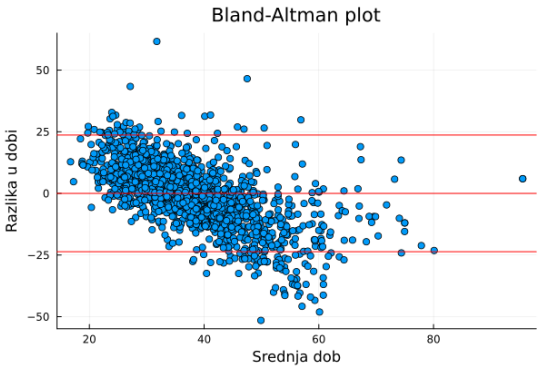
Rezultati linearne regresije, prikazani u tablici 3, pokazuju da je koeficijent za predviđenu dob 1.0, što ukazuje na savršen linearni odnos između predviđene i stvarne dobi. Međutim, *RMSE* (*Root Mean Squared Error*) od 12.07 godina te *MAE* (*Mean Absolute Error*) od 9.12 godina pokazuju da su u prosjeku predikcije odstupale za oko 9 godina, što je značajno odstupanje u kontekstu preciznosti predviđanja.

Distribucija reziduala (razlika između predviđene i stvarne dobi) prikazana je na slici 20, gdje se vidi da većina pogrešaka kreće oko nule, no postoje i značajna odstupanja na obje strane, što može upućivati na potencijalne probleme s modelom za određene dobne skupine.

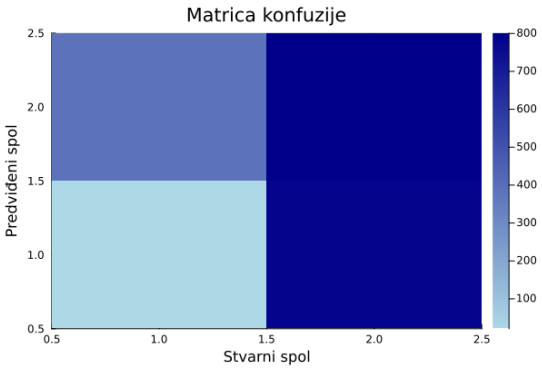
Varijabla	Koeficijent	Standardna greška	t-vrijednost	p-vrijednost
(Intercept)	-1.19e-12	1.41	-0.00	1.000
Predviđena dob	1.00	0.037	27.06	$< 1e-99$

Tablica 3: Rezultati linearne regresije za predviđenu i stvarnu dob

Kako bi se dodatno procijenila točnost predikcija dobi, kreiran je Bland-Altman graf prikazan na slici 21, koji prikazuje srednju dob u odnosu na razliku između predviđene i stvarne dobi i ukazuje na generalno dobru konzistentnost između predviđene i stvarne dobi.



Slika 21: Bland-Altman graf predviđene i stvarne dobi



Slika 22: Matrica konfuzije za predikciju spola

Evaluacija klasifikacije spola provedena je kroz analizu matrice konfuzije, prikazane na slici 22. Matrica konfuzije omogućuje procjenu uspješnosti klasifikacije kroz metrike poput točnosti, preciznosti, osjetljivosti (*recall*) i F1-mjere. S obzirom na rezultate, model postiže točnost od 73.22%, što je solidan rezultat. F1-mjera iznosi 0.78, što ukazuje na uravnoteženost između preciznosti i osjetljivosti.

Na kraju, evaluacija modela završava izračunom *AUC* (*Area Under Curve*) za *ROC* krivulju, koja iznosi 0.740, što ukazuje na dobru sposobnost modela da razlikuje spolove.

5.1.3. Analiza robusnosti na napade

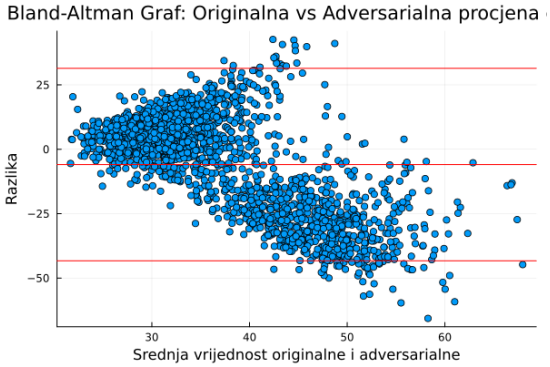
Robusnost modela na *adversarial* napade ključna je za ocjenu otpornosti modela na male promjene ulaznih podataka, koje su često teško uočljive ljudskom oku, ali mogu značajno utjecati na predikcije modela. U ovoj analizi, model je evaluiran na temelju razlika između originalnih predikcija (dob i spol) i predikcija generiranih nakon primjene *adversarial* napada.

Tablica 4 prikazuje osnovne statističke mjere za originalne i *adversarial* predikcije dobi i spola. Primijećena je značajna razlika između originalnih i *adversarial* predikcija, posebice kod predikcija dobi, što ukazuje na osjetljivost modela na napade.

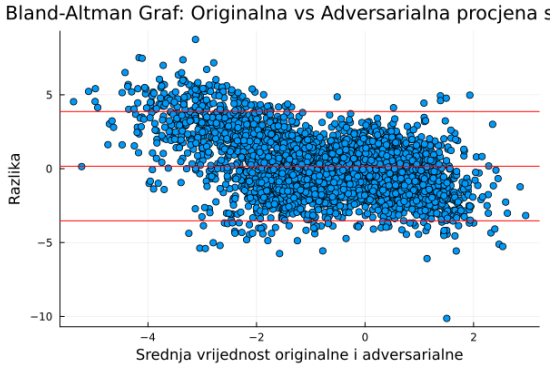
Varijabla	Prosjek	Minimum	Medijan	Maksimum	Ned. vrijednosti
Originalna procj. dobi	34.28	17.77	69.26	91.10	0
Adversarialna procj. dobi	40.22	11.72	91.10	91.10	0
Originalna procj. spola	-1.36	-5.74	1.36	4.42	0
Adversarialna procj. spola	-1.79	-7.92	6.57	6.57	0

Tablica 4: Osnovna statistika za originalne i *adversarial* predikcije dobi i spola

Za procjenu robusnosti, korišteni su Bland-Altman grafovi prikazani na slikama 23 i 24. Ovi grafovi prikazuju razlike između originalnih i *adversarial* predikcija u odnosu na srednju vrijednost. Kod predikcija dobi, primijećena je značajna razlika, s prosječnom razlikom od -5.93 godina, dok je kod predikcija spola zabilježena manja, ali i dalje značajna razlika (0.17).



Slika 23: Bland-Altman graf: Originalna vs adversarijalna procjena dobi



Slika 24: Bland-Altman graf: Originalna vs adversarijalna procjena spola

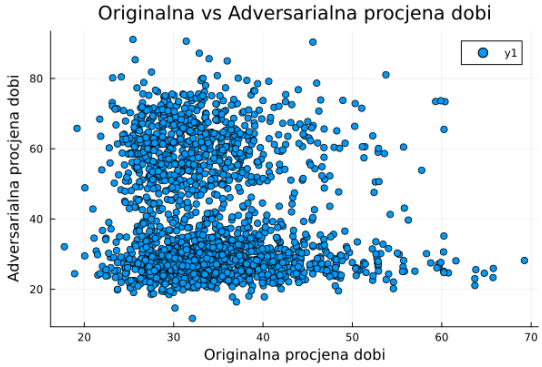
Kako bi se dodatno procijenila osjetljivost modela na napade, provedena je regresijska analiza za originalne i *adversarial* predikcije, čiji su rezultati prikazani u tablici 5. Regresijska analiza za procjene dobi pokazala je negativan koeficijent (-0.24), što ukazuje na smanjenje točnosti predikcija nakon napada. Slično, regresijska analiza za procjene spola pokazuje umjereno pozitivan odnos (0.53) između originalnih i *adversarial* predikcija, ali s velikom varijacijom.

Vizualizacije predikcija prije i nakon *adversarial* napada prikazane su na slikama 25 i 26. Graf raspršenosti za dob pokazuje značajna odstupanja, dok je graf za spol pokazao manju razliku između predikcija. Korelacija između originalnih i *adversarial* predikcija spola iznosi 0.34, što ukazuje na umjereno dobru povezanost, dok je korelacija za predikcije dobi izuzetno niska -0.10, što sugerira visoku osjetljivost modela na napade prilikom predikcije dobi.

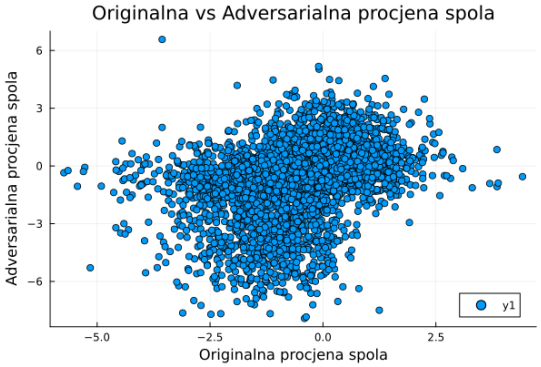
Varijabla	Koeficijent	Standardna greška	t-vrijednost	p-vrijednost
Adversarial dob	-0.24	0.053	-4.55	$< 1e-05$
Adversarial spol	0.53	0.023	22.90	$< 1e-99$

Tablica 5: Rezultati regresijske analize za *adversarial* predikcije dobi i spola

Rezultati ove analize pokazuju da model pokazuje određenu robusnost na *adversarial* napade kod klasifikacije spola, no predikcija dobi je znatno osjetljivija na takve napade, što bi moglo ukazivati na potrebu za dodatnim metodama zaštite od *adversarial* napada tijekom treniranja i testiranja modela.



Slika 25: Originalna vs adversarijalna procjena dobi



Slika 26: Originalna vs adversarijalna procjena spola

5.1.4. Procjena stabilnosti modela

Procjena stabilnosti modela ključna je za razumijevanje kako model reagira na različite perturbacije u ulaznim podacima. U ovoj analizi, model je evaluiran pod različitim scenarijima: dodavanjem šuma, rotacijom, promjenama kontrasta i osvjetljenja. Cilj analize je utvrditi koliko model ostaje stabilan u svojim predikcijama dobi i spola pod ovim uvjetima te u kojoj mjeri perturbacije utječu na točnost modela.

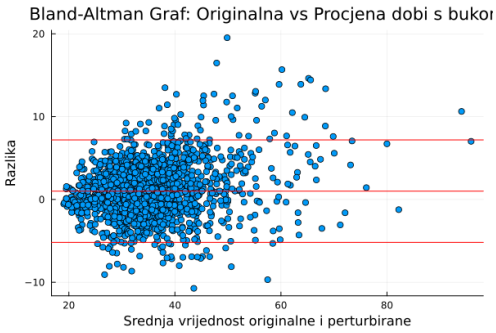
Tablica 6 prikazuje osnovne statističke mjere za originalne predikcije i predikcije dobivene nakon primjene različitih vrsta perturbacija. Vidljivo je da su promjene u predikcijama primjetne kod svih vrsta perturbacija, no najveće promjene bilježe se kod procjena dobi, posebno kod rotacija i promjena kontrasta i osvjetljenja.

Varijabla	**Prosjek**	**Minimum**	**Medijan**	**Maksimum**	**Ned. vrijednosti**
Originalna procjena dobi	36.13	18.76	99.38	99.38	0
Noisy procjena dobi	35.13	18.77	92.36	92.36	0
Rotacija procjena dobi	35.74	24.94	50.61	50.61	0
Brightness procjena dobi	37.12	24.07	71.68	71.68	0
Kontrast procjena dobi	37.04	24.10	70.40	70.40	0
Originalna procjena spola	-1.50	-7.09	0.37	2.42	0
Noisy procjena spola	-1.49	-6.67	0.39	2.34	0
Rotacija procjena spola	-1.51	-2.97	0.41	0.81	0
Brightness procjena spola	-1.46	-4.51	0.32	1.37	0
Kontrast procjena spola	-1.44	-4.44	0.31	1.57	0

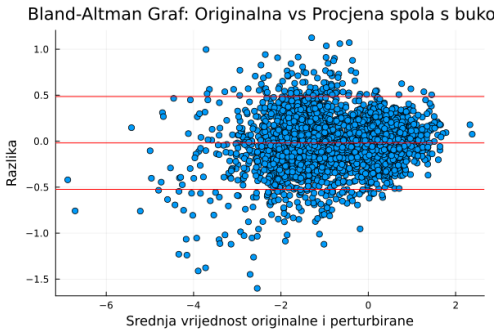
Tablica 6: Osnovna statistika za originalne i perturbirane predikcije dobi i spola

Korelacija između originalnih i perturbiranih predikcija pokazuje koliko su predikcije stabilne pod utjecajem različitih perturbacija. Najveća korelacija primijećena je kod predikcija spola pod šumom ($r = 0.9749$), što sugerira da model ostaje relativno stabilan za ovu varijablu pod šumom. S druge strane, procjene dobi pokazuju veće varijacije, s najnižom korelacijom kod rotacije ($r = 0.1652$), što ukazuje na veliku osjetljivost modela na ovaj tip perturbacije. T-test analiza razlika između originalnih i perturbiranih predikcija potvrđuje statistički značajne razlike u većini scenarija. Na primjer, za procjenu dobi pod šumom, T-test rezultat ukazuje na statistički značajnu razliku ($p < 1e-42$), dok je kod predikcija spola pod rotacijom zabilježen manji, ali još uvijek značajan efekt. Ovi rezultati ukazuju na smanjenu robusnost modela prema promjenama na ulazima.

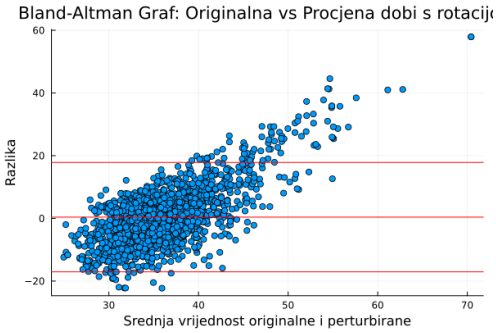
Na slikama 27, 28 i 29, 30 prikazani su Bland-Altman grafovi za razlike između originalnih i perturbiranih predikcija za dob i spol pod šumom i rotacijom. Ovi grafovi prikazuju razlike između procjena i srednjih vrijednosti, te pomažu u identifikaciji sistematskih odstupanja. Vidljivo je da rotacija ima najveći utjecaj na predikcije dobi, s većim odstupanjima od srednjih vrijednosti u usporedbi s drugim perturbacijama. Grafovi raspršenosti na slikama 31 i 32 prikazuju odnos između originalnih i perturbiranih predikcija za dob i spol. Kod predikcija spola pod šumom, primijećeno je relativno usko grupiranje točaka, što ukazuje na veću stabilnost modela u odnosu na perturbaciju. S druge strane, predikcije dobi pod rotacijom pokazuju veća odstupanja i značajno šire raspršenje, što sugerira smanjenje točnosti predikcija.



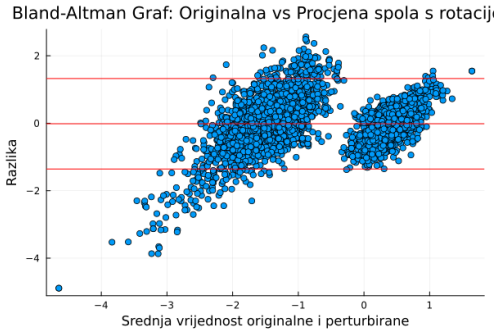
Slika 27: Bland-Altman graf: Originalna vs *Noisy* procjena dobi



Slika 28: Bland-Altman graf: Originalna vs *Noisy* procjena spola

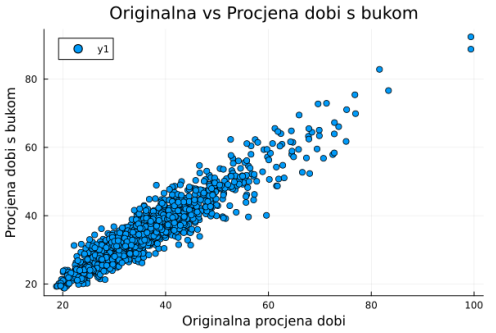


Slika 29: Bland-Altman graf: Originalna vs procjena dobi s rotacijom

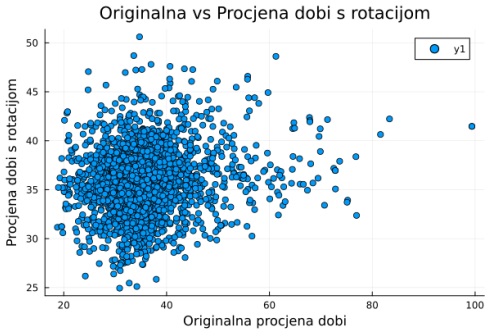


Slika 30: Bland-Altman graf: Originalna vs procjena spola s rotacijom

Ova analiza stabilnosti modela pokazuje da model pokazuje određenu otpornost na perturbacije, posebno pri predikcijama spola. Međutim, predikcije dobi pokazale su osjetljivost na perturbacije, posebno kod rotacija i promjena u kontrastu i osvjetljenju. U budućim iteracijama modela, moguće je razmotriti dodatne tehnike regularizacije i augmentacije podataka kako bi se smanjila osjetljivost na ove promjene i povećala robusnost modela.



Slika 31: Originalna vs *Noisy* procjena dobi



Slika 32: Originalna vs procjena dobi s rotacijom

5.1.5. Testna procjena i usporedba

Testna procjena modela ključan je korak u ocjenjivanju njegovih sposobnosti generalizacije na neviđenim podacima. Kroz ovu analizu procjenjuju se performanse modela na testnom skupu podataka, analiziraju se gubici po epohama te uspoređuje njihovo ponašanje u odnosu na vrijeme treniranja. Ova procjena pruža uvid u to koliko je model stabilan i konzistentan u predikcijama nakon završetka treniranja.

Tablica 7 prikazuje osnovne statističke mjere za testni gubitak i vrijeme trajanja evaluacije po epohama. Srednji testni gubitak iznosi 160.03, dok minimalni gubitak iznosi 143.80 i maksimalni 191.93. Ove varijacije u testnom gubitku ukazuju na to da model još nije postigao potpunu stabilnost tijekom testiranja, no pokazuje generalno prihvatljive performanse.

Varijabla	Prosjek	Minimum	Medijan	Maksimum	Nedostajuće vrijednosti
Epoha	10.5	1	10.5	20	0
Testni gubitak	160.03	143.80	159.64	191.93	0
Vrijeme (s)	4358.76	4217.24	4351.58	4507.18	0

Tablica 7: Osnovna statistika za testne gubitke i vrijeme po epohama

Izračunata korelacija između testnih gubitaka i vremena iznosi $r = -0.7274$, što ukazuje na snažnu negativnu povezanost između gubitka i vremena po epohama. Drugim riječima, kako vrijeme treniranja napreduje, testni gubitak opada, što sugerira da model s vremenom postaje sve bolji u predikcijama i smanjenju gubitaka.

Proveden je T-test kako bi se ispitalo odstupanje srednje vrijednosti testnog gubitka od hipotetičke vrijednosti nula. Rezultati T-testa jasno pokazuju da postoji statistički značajna razlika od nule, s vrlo niskom p-vrijednošću ($p < 1e-22$). Ovi rezultati ukazuju na to da je model postigao konzistentne performanse tijekom testiranja, iako postoji prostor za daljnje smanjenje gubitka.

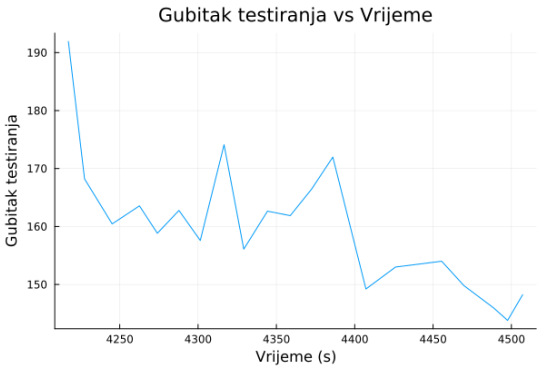
Linearni model koji predviđa testni gubitak po epohama pokazao je značajan negativan koeficijent (-1.37) uz p-vrijednost $p = 0.0004$, što potvrđuje da testni gubitak opada kroz epohe. Ovaj rezultat je potvrđen i kroz vizualizaciju gubitka po epohama, što sugerira da je model u procesu treniranja postajao sve precizniji kako je proces odmicao. Koeficijenti regresijskog modela prikazani su u Tablici 8.

Varijabla	Koeficijent	Standardna greška	t-vrijednost	p-vrijednost
(Intercept)	174.41	3.76	46.36	$< 1e-19$
Epoha	-1.37	0.31	-4.36	0.0004

Tablica 8: Rezultati linearne regresije testnog gubitka kroz epohe

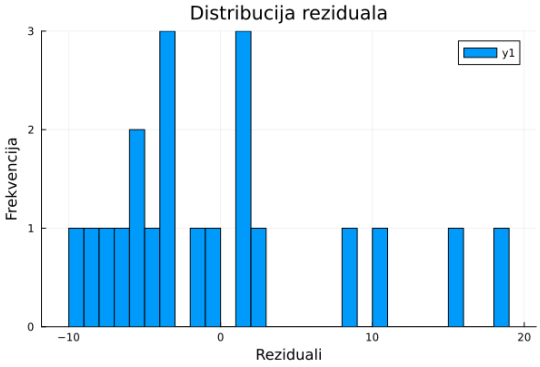
Na slici 33 prikazano je kako testni gubitak opada kroz epohe, dok slika 34 pokazuje odnos između vremena treniranja i gubitka, potvrđujući negativnu korelaciju između ovih dviju varijabli. Jasno je da, kako epohe napreduju, model postiže bolju generalizaciju i smanjuje gubitak.

Slika 33: Testni gubitak kroz epohe

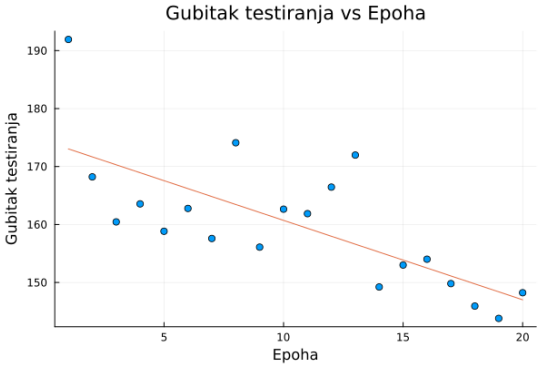


Slika 34: Gubitak u odnosu na vrijeme

Distribucija reziduala na slici 35 također pokazuje kako se model ponaša u odnosu na predikcije. Histogram ukazuje na to da većina reziduala ima vrijednosti blizu nule, što sugerira da model dobro generalizira na testnom skupu, iako postoje neka odstupanja koja bi mogla biti rezultat specifičnih uzoraka ili nedovoljno zastupljenih podataka.



Slika 35: Histogram reziduala na testnom skupu



Slika 36: Graf raspršenosti s regresijskom linijom

Na kraju, graf raspršenosti s regresijskom linijom na slici 36 jasno pokazuje silazni trend, što znači da se gubitak testiranja smanjuje kako broj epoha raste. To sugerira da model poboljšava svoje performanse kako se treniranje odvija. Što je gubitak manji, to su predikcije modela bliže stvarnim vrijednostima.

Ova analiza testne procjene pokazala je da model postiže dobre performanse na testnom skupu podataka, uz stalno smanjenje gubitka kroz epohe. Iako rezultati sugeriraju dobru generalizaciju modela, postoji prostor za daljnje poboljšanje, posebno u smislu smanjenja reziduala i poboljšanja točnosti predikcija na složenijim uzorcima.

5.1.6. Trening performanse i konvergencija

Trening performanse i konvergencija modela ključne su za procjenu koliko učinkovito model uči iz podataka tijekom epoha te koliko brzo gubici opadaju kako treniranje napreduje. Ova analiza se fokusira na promjene u gubicima tijekom treninga, odnos između gubitka i vremena treniranja, kao i sposobnost modela da konvergira prema nižim vrijednostima gubitka kroz epohe.

Tablica 9 prikazuje osnovne statističke mjere za gubitak tijekom treniranja i vrijeme potrebno po epohama. Srednji trening gubitak iznosi 169.49, dok minimalna vrijednost gubitka iznosi 147.66, a maksimalna 214.85. Ove vrijednosti ukazuju na prisutnost stabilnog trenda smanjenja gubitaka, ali s povremenim oscilacijama kroz epohe.

Varijabla	Prosjek	Minimum	Medijan	Maksimum	Nedostajuće vrijednosti
Epoha	10.5	1	10.5	20	0
Trening gubitak	169.49	147.66	169.39	214.85	0
Vrijeme (s)	4358.76	4217.24	4351.58	4507.18	0

Tablica 9: Osnovna statistika za trening gubitke i vrijeme po epohama

Izračunata korelacija između gubitka tijekom treniranja i vremena iznosi $r = -0.918$, što ukazuje na snažnu negativnu povezanost između vremena treniranja i gubitka. Ovaj rezultat jasno pokazuje da model s vremenom postaje sve efikasniji u smanjenju gubitaka, što je ključno za konvergenciju prema stabilnim performansama.

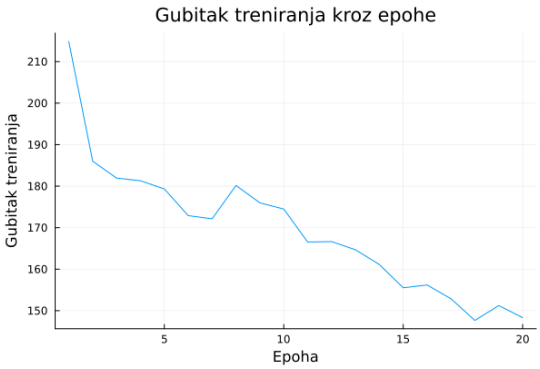
Kako bi se testiralo odstupanje srednje vrijednosti trening gubitka od nule, proveden je T-test. Rezultati T-testa pokazuju značajno odstupanje od nule s p-vrijednošću manjom od $1e{-}20$, što ukazuje na stabilnost modela kroz epohe. Srednja vrijednost trening gubitka značajno je različita od nule, što potvrđuje da model uči i konvergira prema nižim vrijednostima gubitaka.

Linearni model koji predviđa trening gubitak na temelju epoha pokazuje značajan negativan koeficijent (-2.51) uz p-vrijednost $p < 1e-8$, što znači da gubitak opada kako epohe napreduju. Ovaj rezultat je u skladu s očekivanjima, jer kako treniranje napreduje, model postaje sve bolji u predikcijama i smanjenju gubitaka. Rezultati regresije prikazani su u Tablici 10.

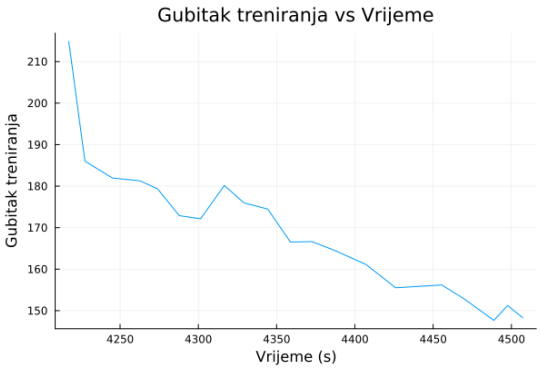
Varijabla	Koeficijent	Standardna greška	t-vrijednost	p-vrijednost
(Intercept)	195.87	2.98	65.64	$< 1e-22$
Epoha	-2.51	0.25	-10.08	$< 1e-08$

Tablica 10: Rezultati linearne regresije trening gubitka kroz epohe

Na slici 37 prikazano je kako trening gubitak opada kroz epohe, dok slika 38 prikazuje odnos između vremena treniranja i gubitka. Ove vizualizacije pokazuju postojan trend smanjenja gubitka, što ukazuje na dobru sposobnost modela da konvergira kroz proces treniranja.

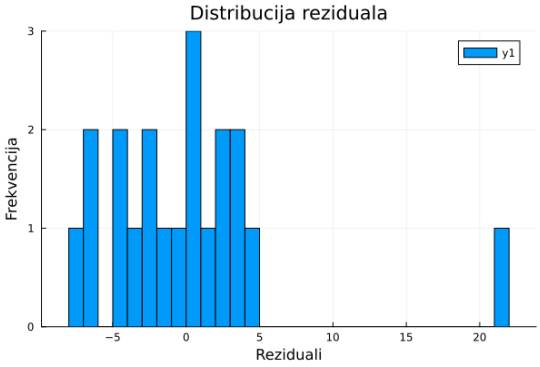


Slika 37: Trening gubitak kroz epohe

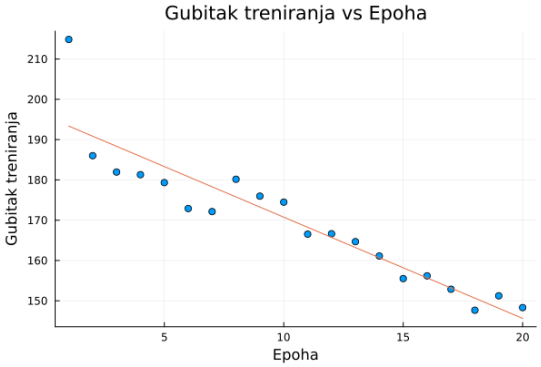


Slika 38: Gubitak u odnosu na vrijeme

Distribucija reziduala na slici 39 dodatno potvrđuje sposobnost modela da postigne stabilnost tijekom treniranja. Histogram reziduala ukazuje na to da većina odstupanja ima vrijednosti blizu nule, što ukazuje na relativno precizne predikcije modela tijekom treniranja, uz manji broj odstupanja u vanjskim regijama.



Slika 39: Histogram reziduala tijekom treniranja



Slika 40: Graf raspršenosti s regresijskom linijom

Graf raspršenosti s regresijskom linijom na slici 40 prikazuje odnos između predviđene i stvarne dobi tijekom treniranja. Regresijska linija ima koeficijent blizu 1, što ukazuje na visoku točnost predikcija modela, iako su vidljiva odstupanja u nekim uzorcima, što ukazuje na potrebu za finim prilagodbama u složenijim uzorcima.

Ova analiza pokazuje da model ima dobre performanse tijekom treniranja te da opadajući trend gubitka kroz epohe jasno ukazuje na konvergenciju modela. Negativna korelacija između gubitka i vremena treniranja dodatno potvrđuje da model postaje učinkovitiji kako epohe napreduju, dok regresijska analiza pokazuje značajan pad gubitaka kroz epohe. Unatoč dobrim rezultatima, prisutnost manjih odstupanja ukazuje na potrebu za daljnjom finom prilagodbom modela kako bi se dodatno poboljšala točnost predikcija.

6. Zaključak

U ovom diplomskom radu istraživali su se različiti aspekti konvolucijskih neuronskih mreža s naglaskom na stabilnost, robusnost i performanse modela prilikom obrade vizualnih podataka. Glavni cilj rada bio je produbiti razumijevanje kako različite ulazne promjene utječu na točnost i pouzdanost CNN-ova, kao i kako se modeli nose s promjenama u ulaznim podacima u stvarnim uvjetima. Tijekom istraživanja korištena je EfficientNet arhitektura kako bi se testiralo prepoznavanje dobi i spola na IMDB-Wiki skupu podataka, dok je provedena analiza kako ulazni šumovi, promjene svjetline, kontrasta, rotacija te adversarijalni napadi utječu na performanse modela.

Kroz rad je detaljno analizirano kako perturbacije poput šuma, promjene svjetline, kontrasta i rotacija utječu na stabilnost modela. Stabilnost CNN-ova ključna je za praktične primjene jer realni scenariji često uključuju nesavršene ili promijenjene ulazne podatke. Modeli koji nisu stabilni skloni su pogreškama koje mogu biti kritične, osobito u osjetljivim područjima poput autonomnih vozila ili medicinske dijagnostike. Rad pokazuje da su perturbacije poput šuma i promjene kontrasta imale relativno nizak utjecaj na predikcije spola, što je vidljivo iz visokih koeficijenata korelacije ($r > 0.88$). Međutim, model se pokazao osjetljivijim na rotacije i promjene u svjetlini kada su u pitanju predikcije dobi, gdje su koeficijenti korelacije bili niži ($r \approx 0.16$ za rotacije), što ukazuje na potrebu za daljnjim prilagodbama modela kako bi se postigla veća stabilnost na ovom tipu zadataka.

Robusnost modela je bila također jedna od ključnih aspekata u istraživanju u ovom radu. Eksperimenti s adversarijalnim napadima pokazali su da se model može značajno destabilizirati čak i malim promjenama u ulaznim podacima. Primjerice, korelacija između originalnih i adversarijalnih predikcija dobi bila je niska ($r = -0.101$), što ukazuje na ranjivost modela prema sofisticiranim napadima. S druge strane, predikcije spola pokazale su se robusnijima na adversarijalne napade, s korelacijom od $r = 0.34$. Ovi rezultati naglašavaju potrebu za dodatnim istraživanjem metoda koje bi model učinile otpornijim na takve napade.

Rezultati evaluacije modela na zadacima prepoznavanja dobi i spola otkrili su da model postiže razumnu točnost u predikcijama spola s točnošću od 0.73, dok je za predikciju dobi koeficijent regresije bio 1.0, što ukazuje na visoku preciznost u predviđanjima dobnih skupina. Međutim, rezidualna analiza i Bland-Altman grafovi ukazali su na postojanje određenih odstupanja i *outlier*-a u podacima koji ukazuju na moguće daljnje prilagodbe modela kako bi se povećala točnost u specifičnim slučajevima.

Detaljno je analiziran proces treniranja modela, uključujući gubitke kroz epohe i odnos između vremena treniranja i gubitka. Gubitak treniranja je postepeno opadao kroz epohe, uz snažnu negativnu korelaciju između gubitka i vremena ($r = -0.918$), što ukazuje na dobru konvergenciju modela. Ovaj rezultat potvrđuje da je EfficientNet arhitektura učinkovita u učenju i stabilizaciji performansi kroz proces treniranja.

Rezultati ovog istraživanja pružaju dubok uvid u ponašanje CNN-a pod različitim uvjetima. Zaključak je da, iako CNN-ovi mogu postići izvanredne rezultate u zadacima prepoznavanja dobi i spola, oni mogu biti i osjetljivi na različite vrste perturbacija i napada. Posebice,

predikcija dobi pokazala se mnogo osjetljivijom na perturbacije i promjene ulaza nego predikcija spola. To je ključni izazov za praktičnu primjenu modela u zadacima koji zahtijevaju visoku točnost i pouzdanost pod nesavršenim uvjetima.

S druge strane, model je pokazao robusnost na određene vrste perturbacija, poput šuma i promjena kontrasta, što upućuje na potencijal za daljnje unaprjeđenje ovih arhitektura. Posebno je važno napomenuti da, iako je model pokazao dobru konvergenciju tijekom treniranja, još uvijek postoji prostor za optimizaciju i smanjenje oscilacija u gubicima tijekom ranih epoha.

Iako je model postigao zadovoljavajuće rezultate u prepoznavanju spola, predikcije dobi bile su osjetljivije na promjene u ulaznim podacima. Buduća istraživanja trebala bi se fokusirati na razvoj metoda koje bi povećale stabilnost predikcija dobi pod različitim uvjetima. Ovo bi moglo uključivati unapređenje arhitekture CNN-a, korištenje tehnika regularizacije ili primjenu specijaliziranih augmentacijskih metoda tijekom treniranja.

Adversarijalni napadi pokazali su se kao značajan izazov za model. Istraživanje metoda obrane od takvih napada trebalo bi biti prioritetno, kako bi se osigurala robusnost modela u stvarnim aplikacijama. Tehnike poput adversarijalnog treniranja ili korištenje obrambenih mehanizama unutar arhitekture mreže mogu pomoći u postizanju ovih ciljeva.

Za autora, ovaj rad predstavlja značajan korak naprijed u razumijevanju stabilnosti i robusnosti CNN-ova u zadacima prepoznavanja dobi i spola, a i općenito. Kroz detaljnu analizu perturbacija, napada i performansi modela, identificirani su ključni izazovi s kojima se suočavaju moderni modeli i razmotrene su smjernice za njihovo unapređenje. S obzirom na široku primjenu CNN-ova u raznim područjima, daljnji razvoj stabilnijih i robusnijih modela ne samo da će povećati točnost i pouzdanost u realnim aplikacijama, već će omogućiti širenje upotrebe CNN-ova u kritičnim sustavima koji zahtijevaju visoku sigurnost i preciznost.

Popis literature

[1] I. Goodfellow, Y. Bengio i A. Courville, *Deep Learning*. MIT Press, 2016., http://www.deeplearningbook.org.

[2] G. James, D. Witten, T. Hastie, R. Tibshirani i J. Taylor, *An Introduction to Statistical Learning: with Applications in Python* (Springer Texts in Statistics). Springer International Publishing, 2023., ISBN: 9783031387463.

[3] R. Szeliski, *Computer Vision - Algorithms and Applications, Second Edition*, 2nd ed. London; New York: Springer, 2022., 1232 str., ISBN: 9783030343712.

[4] E. I. T. C. ACADEMY. „What is the equation for the max pooling?" European IT Certification Institute. (23. 5. 2024.), adresa: https://eitca.org/artificial-intelligence/eitc-ai-adl-advanced-deep-learning/advanced-computer-vision/convolutional-neural-networks-for-image-recognition/what-is-the-equation-for-the-max-pooling/ (pogledano 31. 8. 2024.).

[5] S. Ioffe i C. Szegedy, „Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015., str. 448–456.

[6] K. Zhang, Z. Zhang, Z. Li i Y. Qiao, „Joint Face Detection and Alignment using Multitask Cascaded Convolutional Networks," *CoRR*, sv. abs/1604.02878, 2016. arXiv: 1604.02878.

[7] G. Levi i T. Hassner, „Age and Gender Classification Using Convolutional Neural Networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, str. 34–42, 2015. DOI: 10.1109/CVPRW.2015.7301352.

[8] G. H. A. Krizhevsky I. Sutskever, „ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, 2012., str. 1097–1105.

[9] A. Z. O. Parkhi A. Vedaldi, „Deep Face Recognition," *Proceedings of the British Machine Vision Conference (BMVC)*, 2015., str. 41.1–41.12.

[10] M. K. Hasan, M. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz i G. M. Lee, „Human Face Detection Techniques: A Comprehensive Review and Future Research Directions," *Electronics*, sv. 10, br. 19, 2021., ISSN: 2079-9292. DOI: 10.3390/electronics10192354.

[11] A. Virmaux i K. Scaman, „Lipschitz regularity of deep neural networks: analysis and efficient estimation," *Advances in Neural Information Processing Systems*, sv. 31, 2018.

[12] S. Mallat, „Understanding Deep Convolutional Networks," *Philosophical Transactions of the Royal Society A*, sv. 374, 2016. DOI: 10.1098/rsta.2015.0203.

[13] C. Szegedy, W. Zaremba, I. Sutskever i dr., „Intriguing Properties of Neural Networks," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[14] „Find a lipschitz constant," Stack Exchange Inc. (12. 3. 2014.), adresa: https://math.stackexchange.com/questions/709425/find-a-lipschitz-constant (pogledano 7. 9. 2024.).

[15] J. Hoffman, D. A. Roberts i L. Fei-Fei, „Robust Learning with Jacobian Regularization," *arXiv preprint arXiv:1907.07482*, 2019.

[16] A. Y. Ng, „Feature selection, L1 vs. L2 regularization, and rotational invariance," *Proceedings of the 21st International Conference on Machine Learning*, str. 78, 2004.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever i R. Salakhutdinov, „Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, sv. 15, br. 56, str. 1929–1958, 2014.

[18] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh i P. McDaniel, „Ensemble Adversarial Training: Attacks and Defenses," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[19] I. J. Goodfellow, J. Shlens i C. Szegedy, „Explaining and Harnessing Adversarial Examples," *arXiv preprint arXiv:1412.6572*, 2015.

[20] H. Zhang, M. Cisse, Y. N. Dauphin i D. Lopez-Paz, „mixup: Beyond Empirical Risk Minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[21] W. Luo, Y. Li, R. Urtasun i R. S. Zemel, „Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," *CoRR*, sv. abs/1701.04128, 2017. arXiv: 1701.04128.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva i A. Torralba, *Object Detectors Emerge in Deep Scene CNNs*, 2015. arXiv: 1412.6856 [cs.CV].

[23] A. Antonov i A. Kogtenkov. „How to confuse antimalware neural networks. adversarial attacks and protection," AO Kaspersky Lab. (23. 6. 2021.), adresa: https://securelist.com/how-to-confuse-antimalware-neural-networks-adversarial-attacks-and-protection/102949/ (pogledano 10. 9. 2024.).

[24] R. Rothe, R. Timofte i L. V. Gool, „Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, sv. 126, br. 2-4, str. 144–157, 2018.

[25] R. Rothe, R. Timofte i L. V. Gool, „DEX: Deep EXpectation of apparent age from a single image," *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 12. 2015.

[26] M. Tan i Q. V. Le, „EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *CoRR*, sv. abs/1905.11946, 2019. arXiv: 1905.11946.

[27] P. Ramachandran, B. Zoph i Q. V. Le, „Searching for Activation Functions," *CoRR*, sv. abs/1710.05941, 2017. arXiv: 1710.05941.

Popis slika

Popis tablica

Popis isječaka koda

Prilozi

1. GitHub repozitorij

Poveznica na *GitHub* repozitorij (`https://github.com/pmatisic/fr`) na kojem se nalaze svi programski kôdovi vezani uz ovaj diplomski rad. (Potrebno je kliknuti na ovaj tekst.)