

Primjena konvolucijskih neuronskih mreža za detekciju i klasifikaciju tumora mozga

Ivoš, Natalija

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:212317>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-08-20**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**PRIMJENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA
DETEKCIJU I KLASIFIKACIJU TUMORA MOZGA**

Rijeka, rujan 2022.

Natalija Ivoš

0069077326

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**PRIMJENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA
DETEKCIJU I KLASIFIKACIJU TUMORA MOZGA**

Mentor: Prof. dr. sc. Zlatan Car

Rijeka, rujan 2022.

Natalija Ivoš

0069077326

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Primjena umjetne inteligencije**
Grana: **2.03.06 automatizacija i robotika**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Natalija Ivoš (0069077326)**
Studij: **Diplomski sveučilišni studij elektrotehnike**
Modul: **Automatika**

Zadatak: **Primjena konvolucijskih neuronskih mreža za detekciju i klasifikaciju tumora mozga/Application of convolutional neural networks for detection and classification of brain tumors**

Opis zadatka:

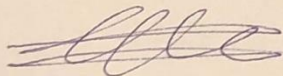
Izvršiti pregled literature iz područja primjene konvolucijskih neuronskih mreža. Razviti modele temeljene na konvolucijskim neuronskim mrežama za detekciju i klasifikaciju tumora mozga. Usporediti dobivene rezultate s dosadašnjim istraživanjima.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Natalija Ivoš

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:



Prof. dr. sc. Zlatan Car

Predsjednik povjerenstva za
diplomski ispit:



Prof. dr. sc. Viktor Sučić

IZJAVA

Sukladno članku 9. Pravilnika o završnom radu, završnom ispitu i završetku sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci izjavljujem da sam samostalno izradila diplomski rad koristeći znanja stečena tijekom studiranja na Tehničkom fakultetu u Rijeci te uz korištenje navedene literature.

Natalija Ivoš

ZAHVALA

Zahvaljujem se profesoru dr.sc Zlatanu Caru, dipl. ing. i stručnom suradniku Danielu Štifaniću mag. ing. el. na pomoći prilikom izrade diplomskog rada.

Sadržaj

1. UVOD	1
2. BAZA PODATAKA	3
2.1. Opis i simptomi tumora mozga	3
2.2. Opis baze podataka	5
3. NEURONSKE MREŽE	7
3.1. Umjetne neuronske mreže.....	7
3.2. Aktivacijske funkcije	10
3.2.1. Sigmoid funkcija.....	10
3.2.2. Hiperbolička tangentna funkcija.....	11
3.2.3. ReLU funkcija	12
3.2.4. Softmax funkcija	13
3.3. Učenje neuronskih mreža.....	14
3.4. Optimizacijski algoritmi	16
3.4.1. Adam	16
3.4.2. SGD	16
3.4.3. RMSprop.....	17
3.5. Konvolucijske neuronske mreže.....	18
3.5.1. Konvolucijski slojevi.....	19
3.5.2. Sloj sažimanja	21
3.5.3. VGG16.....	23
3.5.4. AlexNet	25
3.5.5. ResNet50	26
3.5.6. DenseNet121.....	29
3.5.7. Prilagođena arhitektura.....	31
3.6. Evaluacijske metrike.....	33
4. REZULTATI.....	35
5. ZAKLJUČAK.....	52
6. LITERATURA.....	53
7. POPIS SLIKA.....	56
8. POPIS TABLICA.....	57
9. SAŽETAK.....	58
DODATAK A - Grafovi točnosti i funkcije gubitka za VGG16 model	62
DODATAK B - Grafovi točnosti i funkcije gubitka za AlexNet model	65
DODATAK C - Grafovi točnosti i funkcije gubitka za ResNet50 model	66

DODATAK D - Grafovi točnosti i funkcije gubitka za DenseNet121 model.....	69
DODATAK E - Grafovi točnosti i funkcije gubitka za prilagođeni model.....	71

1. UVOD

Ljudski mozak je središnji organ živčanog sustava koji obavlja svakodnevne životne aktivnosti. Osjetilni organi šalju signale, tj. podražaje mozgu kako bi on vršio obradu i slao izlazne informacije mišićima. Mutacija ili nekontrolirana dioba stanica stvara abnormalne skupine stanica što rezultira tumorom mozga. Tumor je jedan od vodećih uzroka smrti, a tumor mozga je jedan od najsmrtonosnijih. Različite vrste tumora zahtijevaju različite vrste liječenja i pogrešna dijagnoza može uzrokovati smrt pacijenta. Tumori se dijagnosticiraju na temelju njihove veličine, položaja i intenziteta [1]. Složena struktura ljudskog mozga komplicira dijagnozu tumora mozga, stoga se magnetska rezonanca (*eng. Magnetic resonance imaging - MRI*) koristi za dobivanje visokokvalitetnih slika. Nakon snimanja MRI mozga mogu se izlučiti značajke za automatsku klasifikaciju tumora. Različite značajke izvučene iz MRI slika ključni su izvori za klasifikaciju tumora. Nakon izdvajanja značajki, različiti modeli temeljeni na umjetnoj inteligenciji mogu se koristiti za klasifikaciju gdje se klasificira tip tumora: gliom, meningeom, hipofiza i bez tumora. Konvolucijske neuronske mreže (*eng. Convolutional Neural Network - CNN*) su najčešće kombinacija izlučivanja značajki i klasifikatora. Značajke se izdvajaju kroz konvolucijske slojeve i klasifikacija se događa korištenjem potpuno povezanih slojeva na temelju izlučenih značajki [2]. Prijenosno učenje (*eng. Transfer learning*) je koncept gdje se težine već treniranog modela na nekom skupu podataka koriste za neku drugu aplikaciju. Na primjer, za problem koji se razmatra u ovom istraživanju korišteni su ponderi iz modela koji je treniran na skupu podataka ImageNet za 1000 klasa. Upotreba unaprijed treniranih modela pomaže u bržoj konvergenciji.

Postoje mnoge studije u literaturi koje koriste različite metode otkrivanja i klasifikacije tumora iz MRI slika mozga. Neki od međunarodnih članaka o detekciji i klasifikaciji tumora mozga korištenjem dubinskog učenja su S. Basheera i M.S.S.Ran koji su predložili metodu za klasifikaciju tumora mozga gdje se tumor inicijalno segmentira iz MRI slike, a zatim klasificira pomoću prethodno trenirane konvolucijske neuronske mreže (CNN) [3]. Prema literaturi, studiju koja klasificira tumore mozga s najboljim rezultatima proveli su Rehman i suradnici [4]. Primijenili su metode prijenosnog učenja (AlexNet, GoogleNet i VGGNet) za potrebe klasifikacije, te su postigli 98.69% uspješnosti klasifikacije s VGGNet-om. Jednu od novijih studija napravili su Chelghoum i suradnici korištenjem metoda prijenosnog učenja [5]. Primijenili su devet arhitektura prijenosa učenja na skupu podataka Figshare i dobili su točnost klasifikacije od 98.71% koristeći VGG16 s 90 epoha.

Ovaj rad predlaže sustav za dijagnozu i klasifikaciju tumora mozga putem MRI slika korištenjem različitih metoda temeljenih na umjetnoj inteligenciji. Unaprijed trenirani modeli korišteni u ovom radu su: VGG16, ResNet50, DenseNet121 te su korišteni AlexNet i prilagođena arhitektura razvijena za potrebe ovog istraživanja gdje su trenirani svi slojevi. Metode su testirane na javno dostupnom skupu podataka pomoću tri različita optimizacijska algoritma (Adam, SGD, RMSprop). Ciljevi koji se postižu ovim radom su: detekcija tumora mozga, klasifikacija triju vrsta tumora: glioma, meningioma, hipofize i bez tumora te korištenje unaprijed treniranih modela za smanjenje korištenih resursa. Glavni doprinosi istraživanja su: ispitivanje učinkovitosti arhitektura temeljenih na konvolucijskim neuronskim mrežama u svrhu pravilne detekcije i klasifikacije tumora mozga, ispitivanje utjecaja optimizacijskih algoritama na performanse modela te prilagodba postojećih arhitektura u svrhu dodatnog poboljšanja učinkovitosti modela.

2. BAZA PODATAKA

Tumori mozga su najčešća i najagresivnija bolest koja dovodi do vrlo kratkog životnog vijeka u njihovom najvišem stupnju, stoga je planiranje liječenja ključna faza za poboljšanje kvalitete života pacijenata. Općenito, različite tehnike snimanja kao što su kompjutorizirana tomografija, magnetska rezonancija i ultrazvučna slika koriste se za procjenu tumora u mozgu. Skup podataka je besplatno dostupan kao standardni Kaggle skup podataka, a dobiven je tehnikom magnetske rezonancije koja je posebno poželjna za dobivanje slika moždanog tkiva visoke rezolucije u klasifikaciji tumora mozga.

2.1. Opis i simptomi tumora mozga

Tumor mozga nastaje kada se abnormalne stanice formiraju unutar mozga. Postoje dvije glavne vrste tumora: maligni tumori (kancerogeni) i benigni (ne kancerogeni) tumori [6]. Benigni tumori obično sporo rastu te su manje vjerojatnosti da će se vratiti nakon liječenja ili proširiti na druge dijelove mozga, dok maligni tumori mozga rastu brže od benignih tumora te veće vjerojatnosti da će se vratiti nakon liječenja i proširiti na druge dijelove mozga. Tumori se dalje mogu klasificirati kao primarni tumori koji počinju unutar mozga i sekundarni tumori koji se najčešće šire iz tumora smještenih izvan mozga, poznati kao tumori s metastazama u mozgu [7]. Brzina rasta tumora na mozgu može uvelike varirati. Brzina rasta i lokacija tumora mozga određuju kako će on utjecati na funkciju živčanog sustava. Mogućnosti liječenja tumora mozga ovise o vrsti tumora mozga, njegovoj veličini i položaju [8].

Znakovi i simptomi tumora mozga su široki. Simptomi se javljaju bez obzira je li tumor kancerogen ili ne kancerogen. Primarni i sekundarni tumori mozga imaju slične simptome, ovisno o položaju, veličini i brzini rasta tumora [9]. Na primjer, veći tumori u frontalnom režnju mogu uzrokovati promjene u sposobnosti razmišljanja. Međutim, manji tumor u području kao što je Wernickeovo područje (malo područje odgovorno za razumijevanje jezika) može dovesti do većeg gubitka funkcije [10].

Simptomi uzorkovani tumorima mozga mogu uključivati:

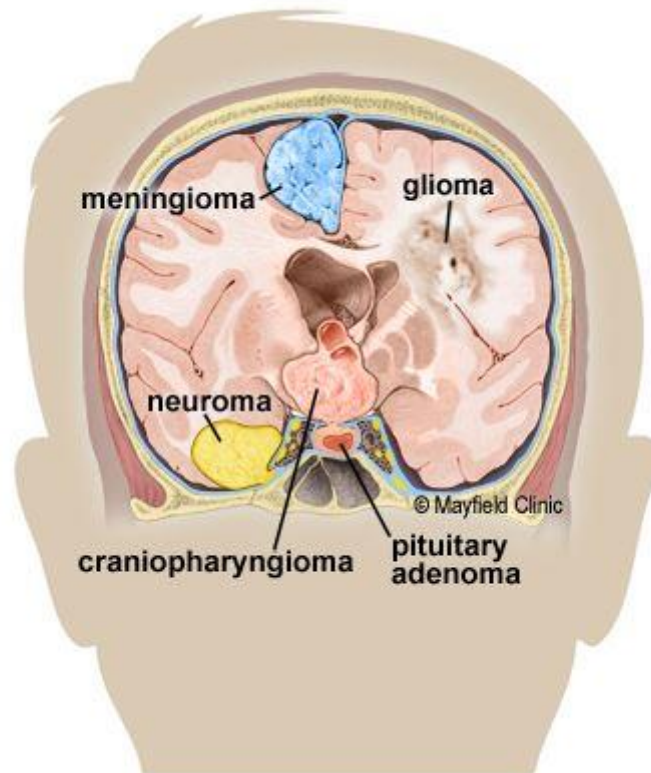
- glavobolje koje postupno postaju sve češće i jače,
- neobjašnjiva mučnina ili povraćanje,
- problemi s vidom poput zamagljenog vida, dvoslike ili gubitka perifernog vida,
- postupni gubitak osjeta ili pokreta u ruci ili nozi,

- poteškoće s ravnotežom,
- govorne poteškoće,
- epilepsija,
- osjećaj jakog umora itd.

Primarni tumori mozga nastaju u samom mozgu ili njemu bliskim tkivima, poput membrane koje prekrivaju mozak (moždanske ovojnice), kranijalni živci, hipofiza ili pinealna žlijezda. Primarni tumori mozga počinju kada normalne stanice razviju promjene (mutacije) u svojoj DNK. DNK stanica sadrži upute koje stanici govore što treba učiniti. Mutacije govore stanicama da brzo rastu i dijele se te da nastave živjeti kada bi zdrave stanice umrle. Rezultat je masa abnormalnih stanica, koja tvori tumor [11].

Postoje mnoge različite vrste primarnih tumora mozga no u ovom radu obrađene su tri vrste koje su prikazane na slici 1. :

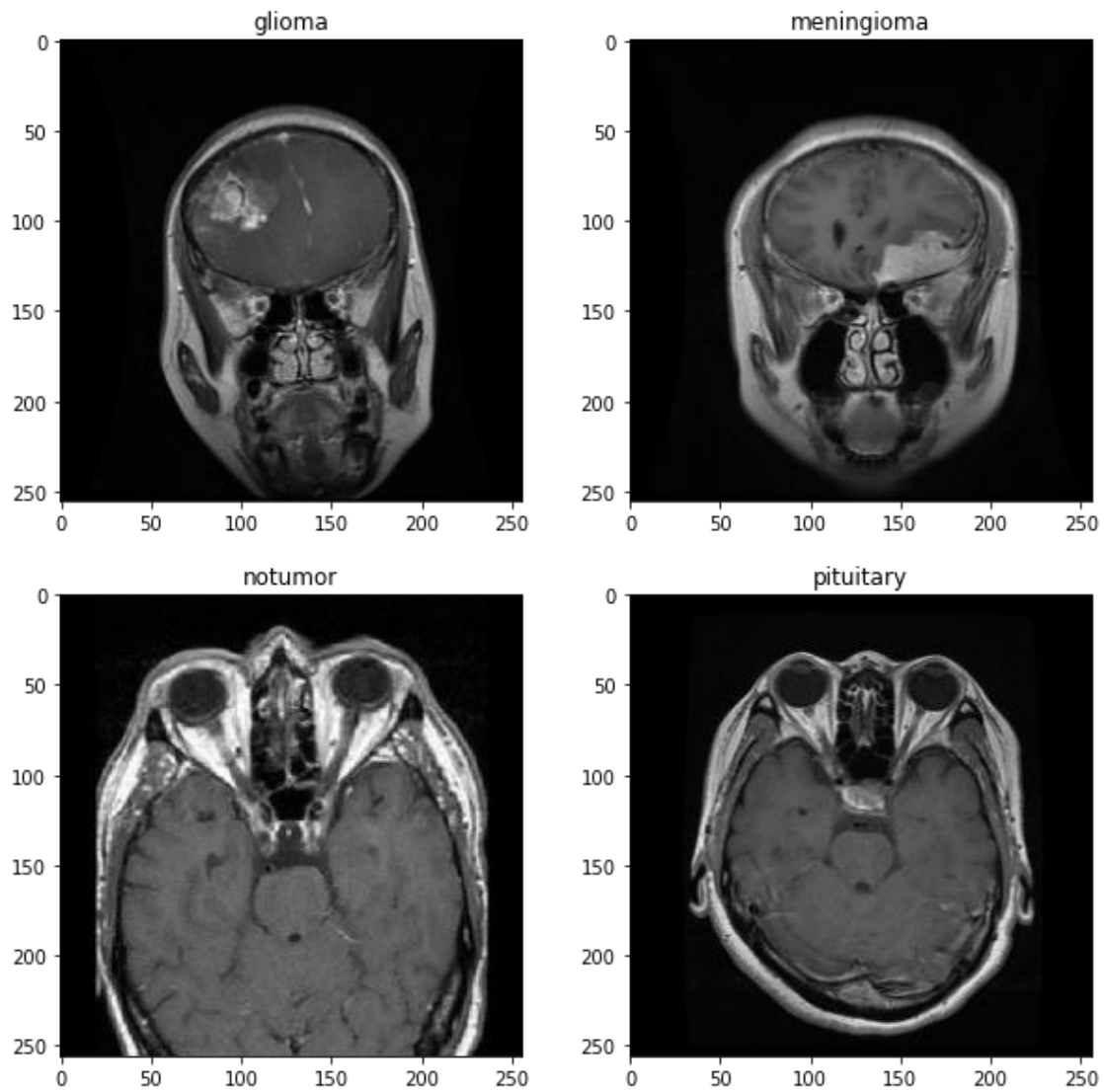
1. Gliom - tumori u središnjem živčanom sustavu (mozak ili leđna moždina) i perifernom živčanom sustavu koji nastaju iz različitih vrsta glijalnih stanica (neuroglia).
2. Meningeom – tumori koji počinju u slojevima tkiva (meninge) koji prekrivaju mozak i leđnu moždinu.
3. Hipofiza – tumor u prednjem dijelu hipofize. Ovo je žlijezda veličine graška koja se nalazi iza hrpta nosa [12].



Slika 1. Lokacija tumora mozga [13]

2.2. Opis baze podataka

Baza podataka kombinacija je tri skupa podataka: Figshare, SARTAJ i Br35H. Slike tumora mozga pohranjuju se kao baze podataka i stvaraju se četiri direktorija, od kojih se svaki sastoji od slika za određenu klasu: glioma, meningeoma, hipofize i bez tumora. Slike su dobivene magnetnom rezonancom koja predstavlja učinkovitu i preciznu dijagnostičku metodu u radiologiji koja nema štetno zračenje za pacijenta. Baza podataka je podijeljena na skupove za treniranje i testiranje/validacija, gdje se 70% podataka koristi u fazi treniranja, a ostatak se koristi u fazi testiranja/validacije. Slika 2. prikazuje primjer tumora mozga za svaku klasu.



Slika 2. Primjer tumora mozga za svaku klasu

Sve slike su digitalizirane u rezoluciji od 256×256 piksela. Tablica 1. prikazuje broj slika svake klase u skupu podataka.

Tablica 1. Distribucija baze podataka u četiri klase

Tip tumora	Broj slika
Gliom	1621
Meningeom	1645
Hipofiza	1999
Nema tumora	1757
Ukupno:	7022

3. NEURONSKE MREŽE

Neuronske mreže su računalni sustavi s međusobno povezanim čvorovima koji rade poput neurona u ljudskom mozgu. Koristeći algoritme neuronske mreže mogu prepoznati skrivene obrasce i korelacije u neobrađenim podacima, grupirati ih i klasificirati te tijekom vremena neprestano učiti i poboljšavati se [14, 15]. Prvu neuronsku mrežu osmislili su Warren McCulloch i Walter Pitts 1943. godine gdje su opisali temeljni rad o tome kako neuroni mogu funkcionirati i modelirali svoje ideje stvaranjem jednostavne neuronske mreže pomoću električnih krugova. Zatim je 1975. godine Kunihiro Fukushima razvio prvu pravu, višeslojnu neuronsku mrežu [16, 17].

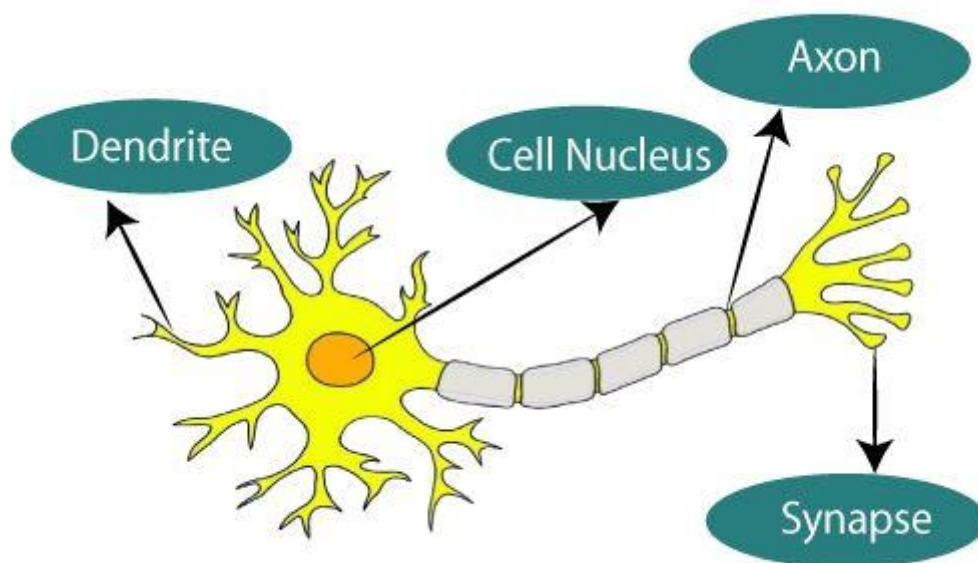
Primarni cilj pristupa neuronske mreže bio je stvoriti računalni sustav koji bi mogao rješavati probleme poput ljudskog mozga. Međutim, s vremenom su istraživači pomaknuli svoj fokus na korištenje neuronskih mreža za rješavanje određenih zadataka, što je dovelo do odstupanja od strogo biološkog pristupa. Od tada neuronske mreže podržavaju različite zadatke, uključujući računalni vid, prepoznavanje govora, strojno prevođenje, filtriranje društvenih mreža, igranje društvenih igara te medicinsku dijagnozu. Kako su se veličine strukturiranih i nestrukturiranih podataka povećavale na razine velikih podataka, ljudi su razvili sustave dubokog učenja, koji su u biti neuronske mreže s mnogo slojeva. Duboko učenje omogućuje hvatanje i rudarenje više i većih podataka, uključujući nestrukturirane podatke.

Postoje tri važne vrste neuronskih mreža gdje svaka ima svoje prednosti i nedostatke [18]:

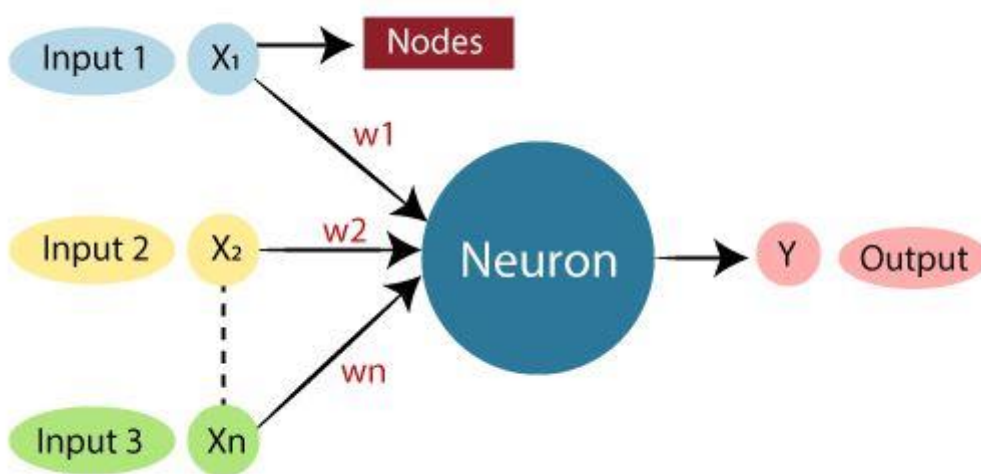
- umjetne neuronske mreže (ANN),
- konvolucijske neuronske mreže (CNN) i
- ponavljajuće neuronske mreže (RNN).

3.1. Umjetne neuronske mreže

Umjetna neuronska mreža (*eng. Artificial Neural Network - ANN*) izvedena je iz bioloških neuronskih mreža koje razvijaju strukturu ljudskog mozga. Slično ljudskom mozgu koji ima neurone međusobno povezane, umjetne neuronske mreže također imaju neurone koji su međusobno povezani u različitim slojevima mreže. Neuroni su poznati kao čvorovi. Na slici 3. prikazana je tipična biološka neuronska mreža, a na slici 4. prikazana tipična umjetna neuronska mreža.



Slika 3. Biološka neuronske mreža [19]



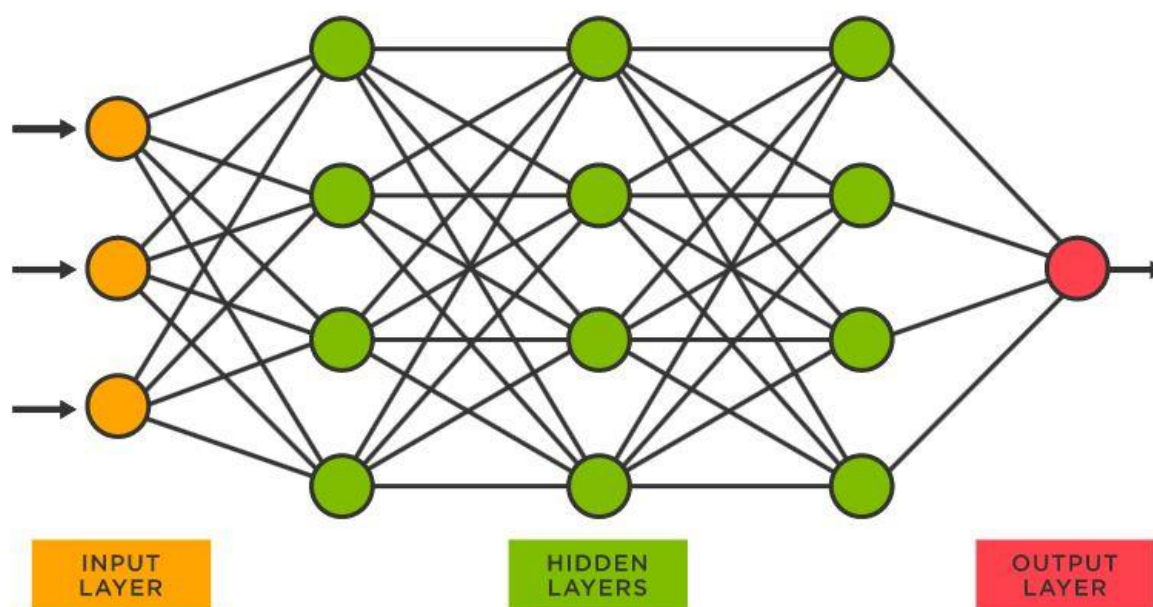
Slika 4. Umjetna neuronska mreža [19]

Biološka neuronska mreža građena je od stvarnih bioloških neurona povezanih u perifernom živčanom sustavu, dok je umjetna neuronska mreža građena od međusobno povezanih umjetnih neurona. U tablici 2. prikazan je odnos između biološke neuronske mreže i umjetne neuronske mreže.

Tablica 2. Odnos biološke neuronske mreže i umjetne neuronske mreže

Biološka neuronska mreža	Umjetna neuronska mreža
Dendrit	Ulaz
Stanična jezgra	Čvorovi
Sinapsa	Težina
Akson	Izlaz

Jedan perceptron ili neuron može se zamisliti kao logistička regresija. Umjetna neuronska mreža skupina je više perceptrona/neurona na svakom sloju. ANN je također poznat kao Feed-Forward neuronska mreža jer se ulazi obrađuju samo u smjeru naprijed. Arhitektura umjetne neuronske mreže prikazana je na slici 5.



Slika 5. Arhitektura umjetne neuronske mreže [20]

ANN se sastoji od 3 osnovna sloja: ulaznog, skrivenog i izlaznog. Ulazni sloj prihvaća ulaze, pruža informacije iz vanjskog svijeta mreži te ne izvodi nikakva izračunavanja, nego čvorovi samo prosljeđuju informacije (značajke) skrivenim slojevima. Čvorovi skrivenih slojeva nisu izloženi vanjskom svijetu, oni su dio apstrakcije koju pruža bilo koja neuronska mreža. Skriveni sloj izvodi sve vrste izračuna na značajkama unesenim kroz ulazni sloj i prenosi rezultat na izlazni sloj. Izlazni sloj estimira rezultat te prenosi informacije koje je mreža naučila u vanjski

svijet. U biti, svaki sloj pokušava naučiti određene značajke. Koristi se za rješavanje problema koji se odnose na tablične podatke, podatke o slici i tekstualne podatke.

Umjetna neuronska mreža sposobna je naučiti bilo koju nelinearnu funkciju, stoga su te mreže popularno poznate kao univerzalni aproksimatori funkcija. ANN ima sposobnost naučiti težine koje mapiraju bilo koji ulaz u izlaz. Jedan od glavnih razloga iza univerzalne aproksimacije je aktivacijska funkcija o kojoj će biti riječ u sljedećem poglavlju [21].

3.2. Aktivacijske funkcije

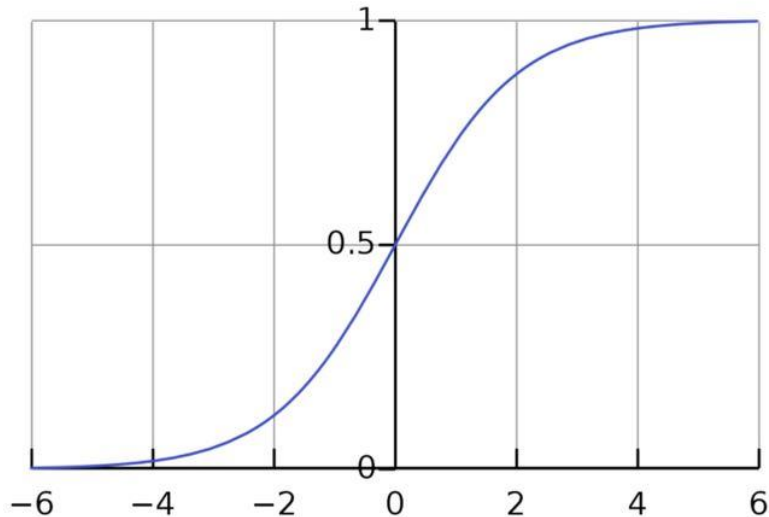
Aktivacijska funkcija je funkcija koja se dodaje u umjetnu neuronsku mrežu kako bi pomogla neuronskim mrežama da shvate komplicirane, visokodimenzionalne i nelinearne skupove velikih podataka. U usporedbi s modelom temeljenim na neuronima koji se nalazi u našem mozgu, funkcija aktivacije na kraju odlučuje što će se poslati sljedećem neuronu. To je točno ono što aktivacijska funkcija također radi u ANN-u. On prima izlazni signal iz prethodne ćelije i pretvara ga u neki oblik koji se može uzeti kao ulaz u sljedeću ćeliju. Neke od najpoznatijih nelinearnih aktivacijskih funkcija su: sigmoid, tangens hiperbolni, ReLU i softmax [22, 23].

3.2.1. Sigmoid funkcija

Sigmoid je nelinearna aktivacijska funkcija, poznata i kao logistička funkcija oblika (3.1.):

$$\sigma(x) = \frac{1}{1 + e^{-ax}} . \quad (3.1.)$$

Izlaz je normaliziran u rasponu od 0 do 1. Diferencijabilan je i daje glatku krivulju gradijenta. Sigmoid se uglavnom koristi prije izlaznog sloja u binarnoj klasifikaciji. Graf sigmoidne funkcije prikazan je na slici 6.



Slika 6. Sigmoidna funkcija [21]

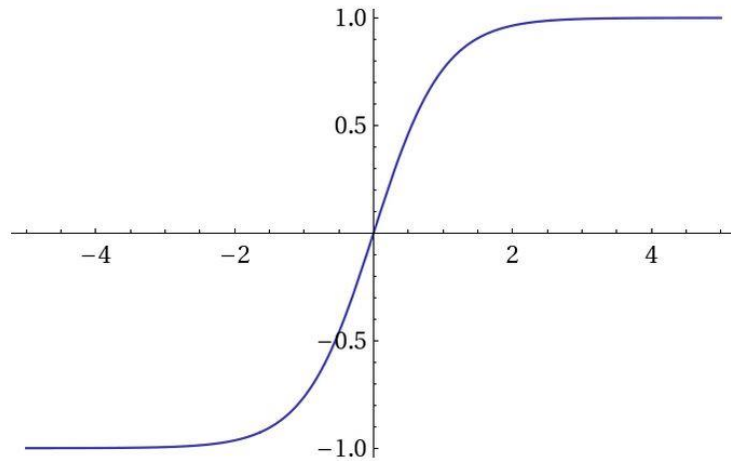
Nedostatak sigmoidne funkcije je problem s nestajanjem gradijenta.

3.2.2. Hiperbolička tangentna funkcija

Hiperbolička tangentna funkcija ili tanh je funkcija s nultim središtem i njezin raspon leži između -1 do 1. Definirana je izrazom (3.2.) kao:

$$f(x) = \frac{1}{1 + e^{-ax}} - 1. \quad (3.2.)$$

Kako je ova funkcija centrirana na nulu, to olakšava modeliranje ulaza koji imaju izrazito negativne, neutralne i jako pozitivne vrijednosti. Gradijenti za tanh su strmiji od sigmoidnog. Uglavnom se koristi u binarnoj klasifikaciji. Graf tanh funkcije prikazan je na slici 7.



Slika 7. Hiperbolička tangentna funkcija [21]

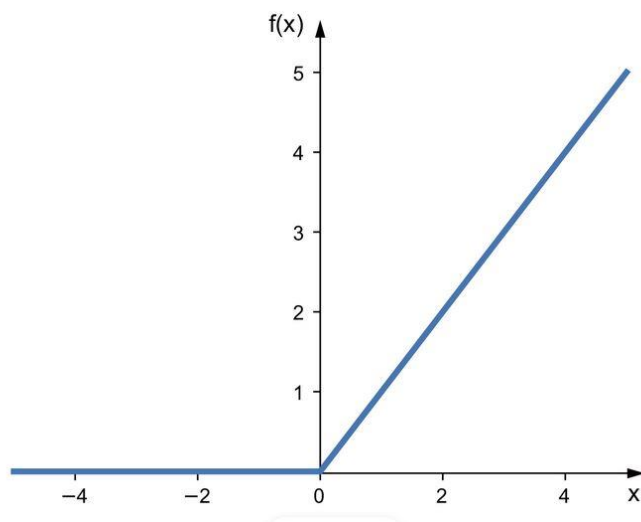
Nedostatak tanh funkcije je problem s nestajanjem gradijenta.

3.2.3. ReLU funkcija

ReLU (*eng. Rectified Linear Unit*) najčešće je korištena funkcija aktivacije u skrivenim slojevima modela dubokog učenja definirana izrazom (3.3.) kao:

$$f(x) = \max(0, x) . \quad (3.3.)$$

Pruža prednost u izvedbi i generalizaciji u usporedbi sa sigmoidnim i tanh aktivacijskim funkcijama. Uz povećanu ukupnu brzinu računanja, ReLU pruža brže računanje budući da ne izračunava eksponencijale i dijeljenja. Graf ReLU funkcije prikazan je na slici 8.



Slika 8. ReLU funkcija [21]

Nedostatak ReLU funkcije je problem mrtve aktivacije koja se događa kada je vrijednost 0. Tijekom procesa unazadne propagacije, težine za neke neurone nisu ažurirane, što može stvoriti mrtve neurone koji se nikad ne aktiviraju. Sve negativne ulazne vrijednosti odmah postaju nula što smanjuje sposobnost modela da se pravilno prilagodi ili trenira iz podataka. Koristi se najčešće u skrivenim slojevima.

3.2.4. Softmax funkcija

Funkcija softmax je još jedna vrsta aktivacijske funkcije koja se koristi u neuronskim mrežama za izračunavanje distribucije vjerojatnosti iz vektora realnih brojeva. Ova funkcija generira izlaz koji se kreće između vrijednosti 0 i 1 i sa zbrojem vjerojatnosti jednakim 1. Funkcija softmax predstavljena je izrazom (3.4.) na sljedeći način:

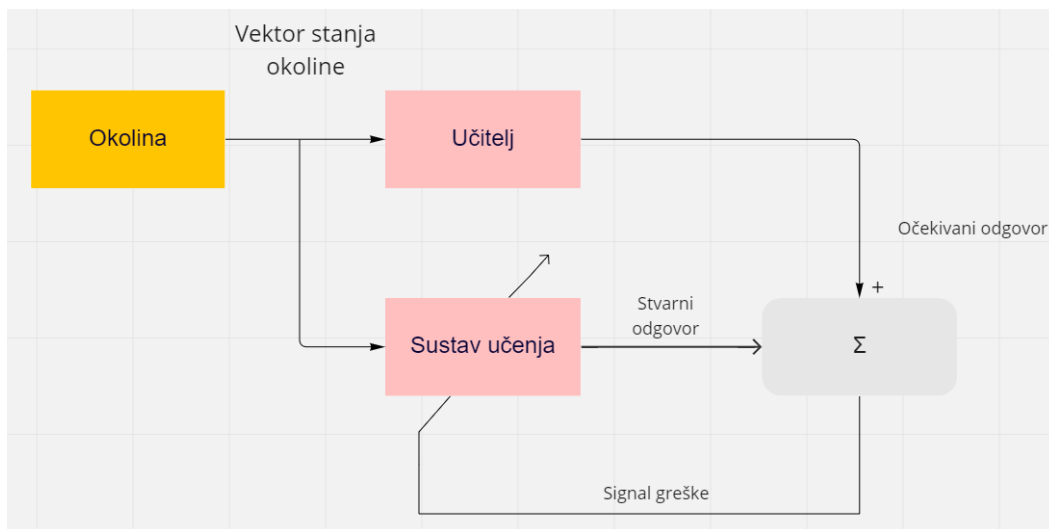
$$f(x_i) = \frac{\exp(x_i)}{\exp \sum(x_j)} . \quad (3.4.)$$

Ova se funkcija uglavnom koristi u modelima s više klasa gdje vraća vjerojatnosti svake klase, pri čemu ciljna klasa ima najveću vjerojatnost. Pojavljuje se u gotovo svim izlaznim slojevima ako je problem klasifikacijski. Primarna razlika između sigmoidne i softmax aktivacijske funkcije je u tome što se sigmoidna koristi u binarnoj klasifikaciji, dok se softmax koristi za višeklasnu klasifikaciju [25].

3.3. Učenje neuronskih mreža

Strojno učenje je potpodručje umjetne inteligencije (*eng. Artificial Intelligence - AI*) koje sustavima omogućuje automatsko učenje i poboljšanje iz iskustva. Za proces učenja treba imati na raspolaganju neka opažanja ili podatke kako bi se istražili potencijalni temeljni obrasci skriveni u podacima. Obrasci predstavljaju neke funkcije ili granice odlučivanja. Načini učenja se kategoriziraju kao učenje s učiteljem, učenje bez učitelja i ojačano učenje [26].

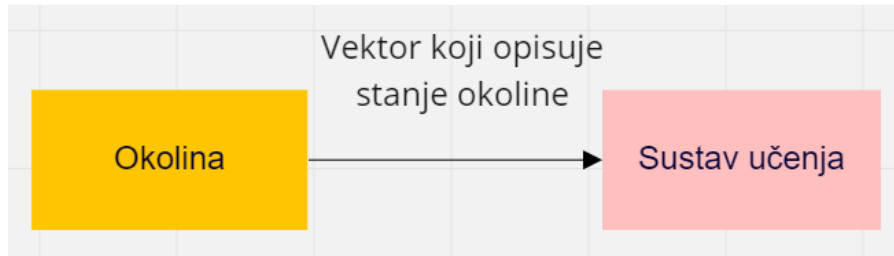
Učenje s učiteljem naziva se još i nadzirano učenje, a kao što naziv sugerira odvija se pod nadzorom učitelja. Ovaj proces učenja je ovisan. Tijekom obuke ANN-a pod nadzorom učenja, ulazni vektor se predstavlja mreži, koja će dati izlazni vektor. Ovaj izlazni vektor se uspoređuje sa željenim izlaznim vektorom. Generira se signal greške ako postoji razlika između stvarnog izlaza i željenog izlaznog vektora. Na temelju ovog signala pogreške, težine se prilagođavaju dok se stvarni izlaz ne uskladi sa željenim izlazom. Postoje dvije glavne vrste problema nadziranog učenja, a to su klasifikacija koja uključuje predviđanje oznake klase i regresija koja uključuje predviđanje numeričke vrijednosti. Blok dijagram učenja s učiteljem prikazan je na slici 9.



Slika 9. Učenje s učiteljem [27]

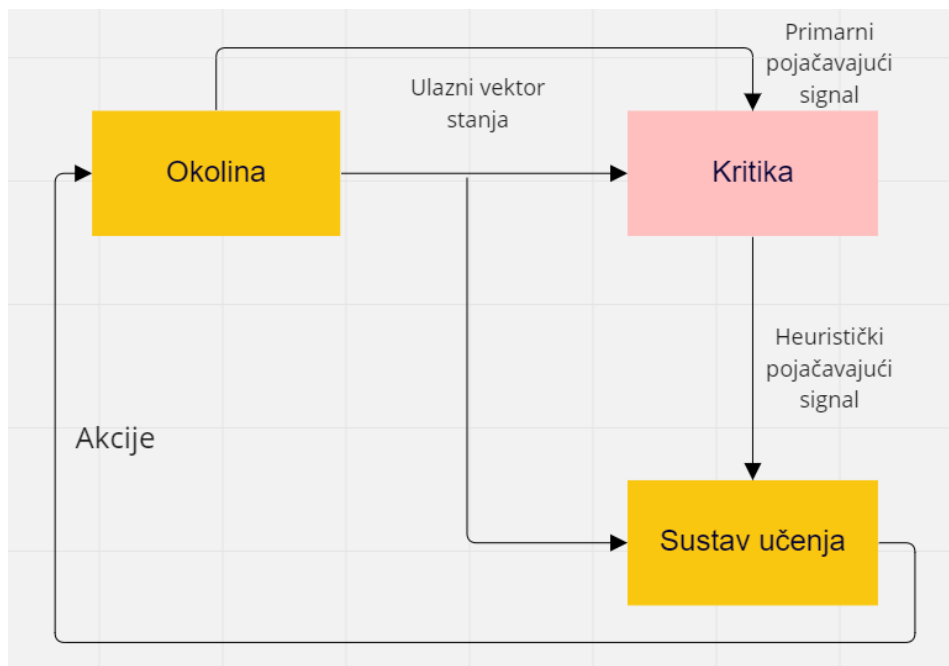
Učenje bez učitelja je vrsta učenja koja se odvija bez nadzora učitelja. Ovaj proces učenja je neovisan. Tijekom obuke ANN-a pod nenadziranim učenjem, ulazni vektori sličnog tipa kombiniraju se u obliku klastera. Kada se primijeni novi ulazni obrazac, tada neuronska mreža daje izlazni odgovor koji označava klasu kojoj ulazni uzorak pripada. Ne postoji povratna

informacija iz okoline o tome što bi trebao biti željeni output i je li točan ili netočan. Stoga, u ovoj vrsti učenja, sama mreža mora otkriti obrasce i značajke iz ulaznih podataka, te odnos između ulaznih podataka i izlaza. Blok dijagram učenja bez učitelja prikazan je na slici 10.



Slika 10. Učenje bez učitelja [27]

Ojačano učenje je vrsta učenja koja se koristi za jačanje mreže oko nekih kritičnih informacija. Ovaj proces učenja sličan je nadziranom učenju, no s puno manje informacija. Tijekom treninga mreže uz ojačano učenje, mreža dobiva neke povratne informacije iz okoline. To je čini donekle sličnim nadziranom učenju. Međutim, povratne informacije dobivene ovdje su evaluativne, a ne instruktivne, što znači da nema učitelja kao kod nadziranog učenja. Nakon primitka povratnih informacija, mreža vrši prilagodbe težina kako bi u budućnosti dobila bolje informacije o kritičarima. Blok dijagram ojačanog učenja prikazan je na slici 11.



Slika 11. Ojačano učenje [27]

3.4. Optimizacijski algoritmi

Za postizanje boljih rezultata mreže koristi se proces optimizacije u kojem se iterativno trenira model koji rezultira maksimalnom i minimalnom procjenom funkcije. Optimizacijski algoritmi predstavljaju metode koje se koriste za promjenu atributa neuronske mreže kao što su težina i pomak kako bi se smanjili gubici te dali što bolji rezultati. Uspoređuju se rezultati u svakoj iteraciji i ažuriraju se gradijenti dok se ne postignu optimalni rezultati i stvori točan model s nižom stopom pogreške.

3.4.1. Adam

Adam (*eng. Adaptive Moment Estimation*) je metoda kod koje se izračunava adaptivna stopa učenja za svaki parametar. Smatra se kao kombinacija RMSpropa i stohastičkog gradijentnog spuštavanja s momentom zato što pohranjuje eksponencijalno opadajući prosjek prošlih kvadratnih gradijenata te čuva eksponencijalno opadajući prosjek prošlih gradijenata. Za procjenu trenutka, Adam koristi eksponencijalno pomične prosjeke gradijenta i kvadrata gradijenta predstavljeno izrazom (3.5.) i (3.6.):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.5.)$$

i

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (3.6.)$$

gdje m_t i v_t predstavljaju procjene prvog trenutka (srednja vrijednost) odnosno drugog trenutka (necentrirana varijanca) gradijenata, a g predstavlja gradijent. Hiperparametri $\beta_1, \beta_2 \in [0, 1)$ kontroliraju eksponencijalne stope opadanja pomičnih prosjeka [28, 29].

3.4.2. SGD

SGD (*eng. Stochastic gradient descent*) je iterativna metoda za optimizaciju objektivne funkcije s odgovarajućim svojstvima glatkoće. Smatra se stohastičkom aproksimacijom optimizacije spuštavanja gradijenta zbog toga što zamjenjuje stvarni gradijent njegovom procjenom. SGD smanjuje vrlo visoko računarsko opterećenje, pružajući brže iteracije za nižu stopu konvergencije. U SGD-u umjesto uzimanja cijelog skupa podataka za svaku iteraciju,

nasumično se odabire serije podataka što znači da se uzima samo nekoliko uzoraka iz skupa podataka prikazano formulom (3.7.).

$$w = w - \eta \nabla Q_i(w), \quad (3.7.)$$

gdje η je stopa učenja koja kontrolira veličinu koraka u prostoru parametara.

Budući da se ne koristi cijeli skup podataka, već njegove skupine za svaku iteraciju, put kojim je prošao algoritam pun je šuma stoga SGD koristi veći broj iteracija za postizanje lokalnih minimuma. Zbog povećanja broja ponavljanja, povećava se ukupno vrijeme izračuna [30].

3.4.3. RMSprop

RMSprop (*eng. Root Mean Square Propagation*) je tehnika optimizacije temeljena na gradijentu koja se koristi u treniranju neuronskih mreža. Pokušava prigušiti oscilacije te uklanja potrebu za podešavanjem stope učenja automatski. Štoviše, RMSprop odabire različitu stopu učenja za svaki parametar. Svako ažuriranje se radi prema jednadžbama (3.8.), (3.9.) i (3.10.) :

$$v_t = \rho v_{t-1} + (1 - \rho) * g_t^2, \quad (3.8.)$$

$$\Delta w_t = -\frac{\eta}{\sqrt{v_t + \epsilon}} * g_t, \quad (3.9.)$$

$$w_{t+1} = w_t + \Delta w_t, \quad (3.10.)$$

te zasebno za svaki parametar, gdje v_t predstavlja eksponencijalni prosjek kvadrata gradijenata, η je stopa učenja, a g_t gradijent u trenutku t duž w^j .

RMSprop koristi pomični prosjek kvadratnih gradijenata za normalizaciju gradijenta. Ta normalizacija uravnotežuje veličinu koraka, smanjujući korak za velike gradijente kako bi se izbjegla eksplozija gradijenata i povećavajući korak za male gradijente kako bi se izbjeglo nestajanje. Jednostavno rečeno, RMSprop koristi prilagodljivu stopu učenja umjesto da stopu učenja tretira kao hiperparametar. To znači da se stopa učenja mijenja tijekom vremena [31].

3.5. Konvolucijske neuronske mreže

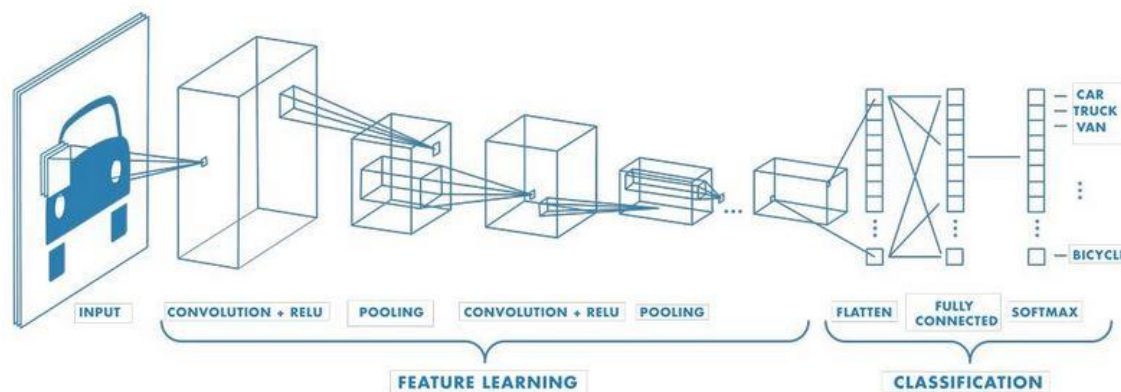
Konvolucijska neuronska mreža (*eng. Convolutional Neural Network-CNN*) algoritam je dubokog učenja koji uči izravno iz podataka, eliminirajući potrebu za ručnim izdvajanjem značajki. CNN je posebno koristan za pronalaženje uzoraka na slikama za prepoznavanje objekata, lica i prizora. Oni također mogu biti vrlo učinkoviti za klasificiranje neslikovnih podataka kao što su audio, vremenski nizovi i signalni podaci. CNN koristi posebnu tehniku koja se zove konvolucija. U matematici konvolucija je matematička operacija na dvije funkcije koja rezultira trećom funkcijom koja izražava kako je oblik jedne modificiran drugom.

Važni čimbenici korištenja CNN-a:

- CNN eliminira potrebu za ručnim izdvajanjem značajki, odnosno uči izravno značajke,
- CNN daje vrlo precizne rezultate prepoznavanja i
- CNN se može prekvalificirati za nove zadatke prepoznavanja, što omogućuje nadogradnju na već postojećim mrežama.

Konvolucijska neuronska mreža može imati nebrojeno slojeva od kojih svaki uči otkrivati različite značajke slike. Filtri se primjenjuju na svaku sliku za treniranje u različitim rezolucijama, a izlaz svake slike koristi se kao ulaz za sljedeći sloj. Filtri mogu početi kao vrlo jednostavne značajke, kao što su svjetlina i rubovi te povećati složenost do značajki koje definiraju objekt. Kao i druge neuronske mreže, CNN se sastoji od ulaznog sloja, izlaznog sloja i puno skrivenih slojeva između. Model dubokog učenja prati veze između ulaznih podataka, više slojeva i izlaza. Ovi slojevi izvode operacije koje mijenjaju podatke s namjerom učenja značajki specifičnih za podatke. Tri najčešća sloja su: konvolucijski sloj, aktivacijska funkcija (najčešće ReLU) i sloj sažimanja.

Konvolucija provlači ulazne slike kroz skup konvolucijskih filtara od kojih svaki aktivira određene značajke slike. Aktivacijska funkcija (ReLU) omogućuje brži i učinkovitiji trening mapiranjem negativnih vrijednosti na nulu i održavanjem pozitivnih vrijednosti. To se ponekad naziva aktivacijom jer se samo aktivirane značajke prenose u sljedeći sloj. Sloj sažimanja pojednostavljuje izlaz izvođenjem nelinearnog smanjivanja uzorkovanja, smanjujući broj parametara koje mreža mora naučiti [32, 33]. Primjer osnovne arhitekture konvolucijske neuronske mreže prikazan je na slici 12.



Slika 12. Primjer osnovne arhitekture konvolucijske neuronske mreže [33]

3.5.1. Konvolucijski slojevi

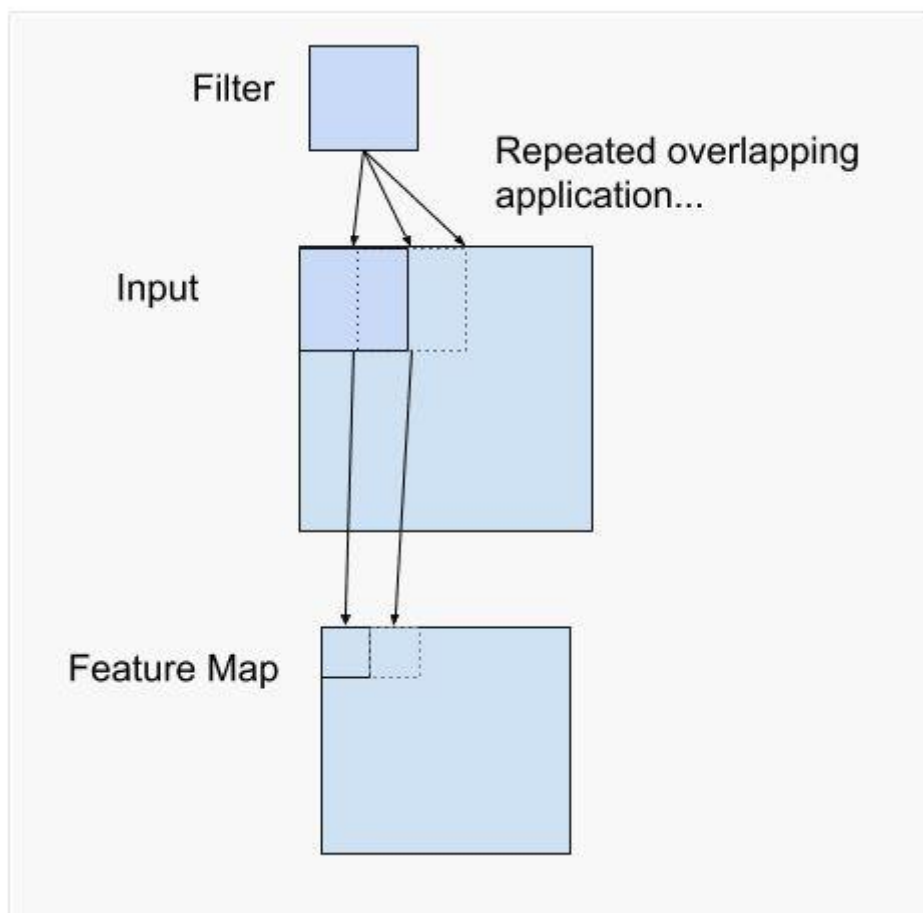
Konvolucijski sloj temelj je CNN-a i na njemu se odvija većina izračuna. Konvolucija je jednostavna primjena filtra na ulaz koja rezultira aktivacijom. Ponovom primjenom istog filtra na ulaz rezultira mapom aktivacija koja se naziva mapa značajki (*eng. feature map*), pokazujući lokacije i intenzitet detektirane značajke na ulazu kao što je slika. Inovacija konvolucijskih neuronskih mreža je sposobnost automatskog paralelnog učenja velikog broja filtara specifičnih za skup podataka za obuku pod ograničenjima specifičnog problema prediktivnog modeliranja kao što je klasifikacija slike. Rezultat daje specifične značajke koje se mogu otkriti bilo gdje na ulaznim slikama.

CNN specijalizirana je vrsta modela neuronske mreže dizajnirana za rad s dvodimenzionalnim slikovnim podacima, iako se mogu koristiti s jednodimenzionalnim i trodimenzionalnim podacima. Konvolucijski sloj predstavlja centralni dio konvolucijske neuronske mreže te izvodi operaciju koja se naziva konvolucija. U kontekstu konvolucijske neuronske mreže, konvolucija je linearna operacija koja uključuje množenje skupa težina s ulazom, slično tradicionalnoj neuronskoj mreži. S obzirom da je tehnika dizajnirana za dvodimenzionalni unos, množenje se izvodi između niza ulaznih podataka i dvodimenzionalnog niza težina, tj. filtara ili kernela. Filtar je manji od ulaznih podataka, a vrsta množenja koja se primjenjuje između dijela ulaza veličine filtra i filtra je točkasti umnožak. Točkasti umnožak je množenje po elementima između veličine filtra ulaza i filtra koji se zatim zbraja što uvijek rezultira jednom vrijednošću. Budući da rezultira jednom vrijednošću, operacija se često naziva skalarni produkt. Namjerno korištenje filtra manjeg od ulaznog omogućuje da se taj isti filtar (skup težina) pomnoži ulaznim

nizom više puta na različitim točkama na ulazu. Filtar se primjenjuje na svaki preklapajući dio ili veličinu filtra ulaznih podataka odozgo prema dolje, odnosno slijeva na desno.

Sustavna primjena istog filtra na sliku je inovativna ideja. Ako je filter dizajniran za otkrivanje određene vrste značajke u ulazu, tada sustavna primjena tog filtra na cijeloj ulaznoj slici omogućuje filtru priliku da otkrije tu značajku bilo gdje na slici. Ta sposobnost se obično naziva invarijantnošću prijevoda. Invarijantnost lokalnog prijevoda je vrlo korisno svojstvo ako je bitnije je li neka značajka prisutna, nego gdje se točno nalazi.

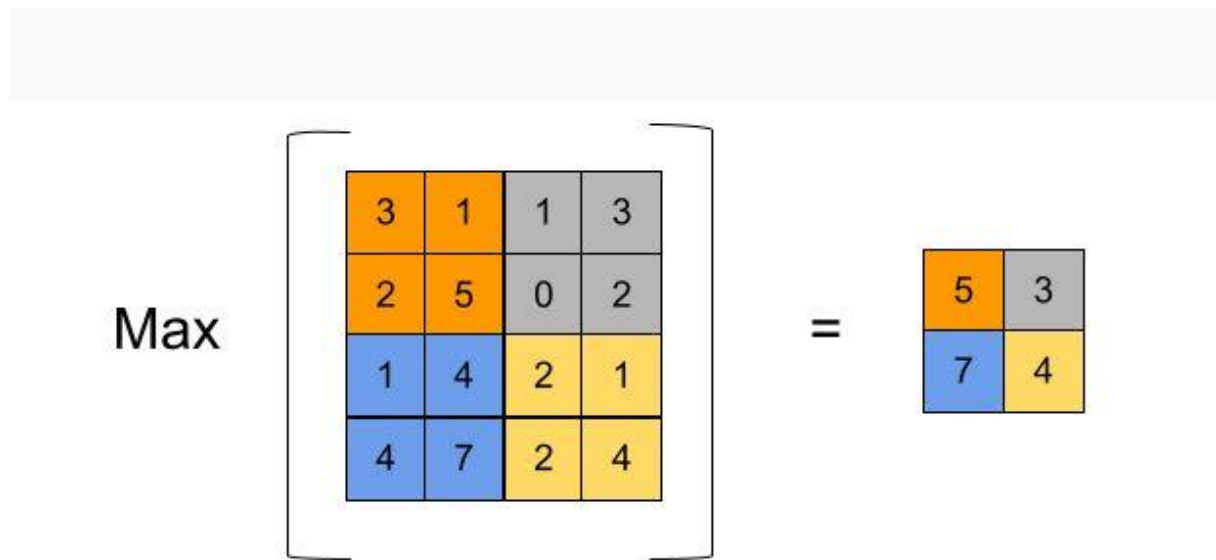
Budući da se filter primjenjuje više puta na ulazni niz, rezultat je dvodimenzionalni niz izlaznih vrijednosti koje predstavljaju filtriranje ulaza, tj. mapu značajki. Jednom kada je mapa značajki stvorena, svaka vrijednost u mapi značajki može proći kroz nelinearnost (npr. ReLU) slično kao što se radi za izlaze potpuno povezanog sloja [34]. Primjer filtra primijenjenog na dvodimenzionalni unos za izradu mape značajki prikazan je na slici 13.



Slika 13. Filtar primijenjen na dvodimenzionalni ulaz za izradu mape značajki [35]

3.5.2. Sloj sažimanja

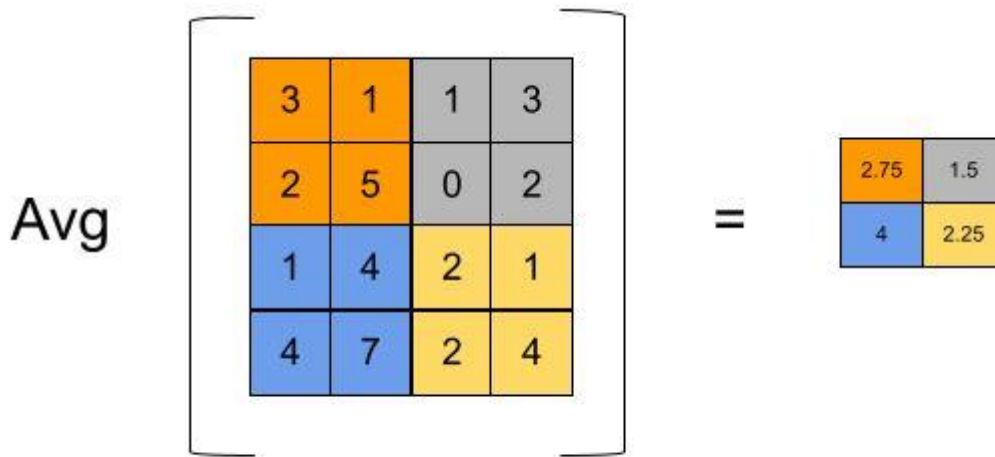
Osnovni postupak sažimanja (*eng. Pooling*) vrlo je sličan operaciji konvolucije. Odabrani filter pomakne se preko izlazne mape značajki prethodnog konvolucijskog sloja. Najčešće korištena veličina filtra je 2×2 i klizi preko ulaza korakom od 2. Na temelju vrste odabrane operacije sažimanja, filter udruživanja izračunava izlaz na receptivnom polju (dio mape značajki ispod filtra). Postoji nekoliko pristupa udruživanju. Najčešće korišteni pristupi su maksimalno sažimanje (*eng. Max Pooling*) i prosječno sažimanje (*eng. Average Pooling*). U maksimalnom sažimanju, filter jednostavno odabire maksimalnu vrijednost piksela u receptivnom polju. Primjer maksimalnog sažimanja prikazan je na slici 14.



Slika 14. Primjer max poolinga [36]

Kao što je vidljivo na slici, ukoliko u polju ima četiri piksela s vrijednostima 3, 1, 2 i 5, odabire se najveći, odnosno 5. Max pooling smanjuje dimenzionalnost slika smanjenjem broja piksela u izlazu iz prethodnog konvolucijskog sloja.

Prosječno sažimanje funkcionira izračunavanjem prosječne vrijednosti piksela u receptivnom polju. Primjer prosječnog sažimanja prikazan je slikom 15.



Slika 15. Primjer average-poolinga [36]

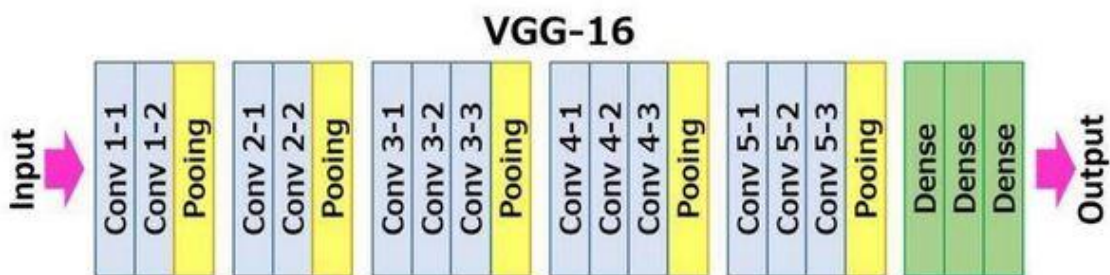
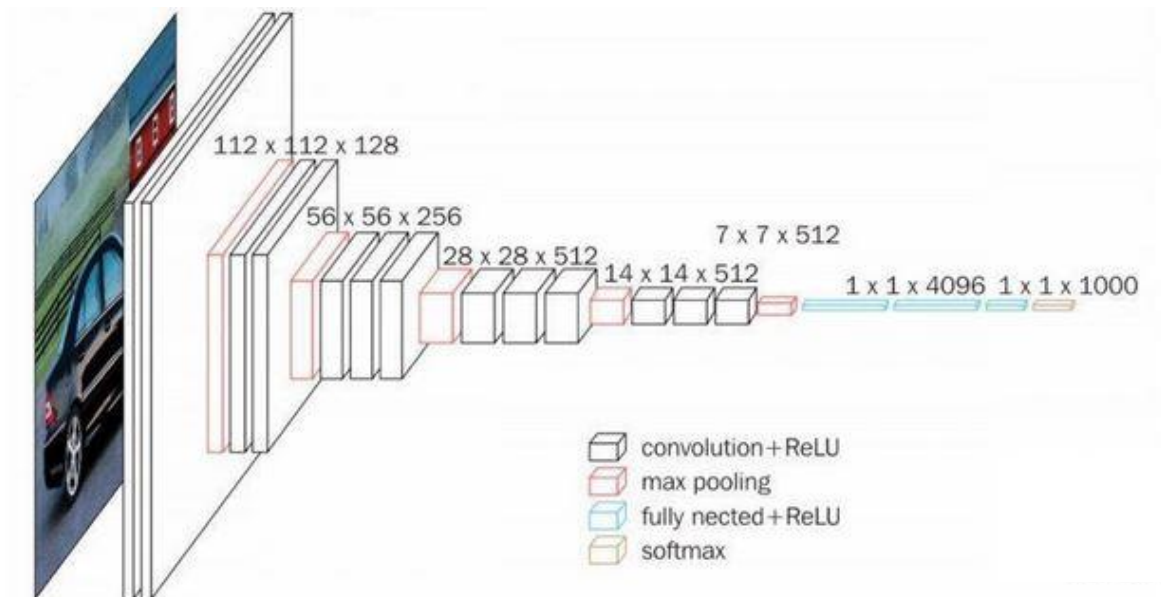
U ovom primjeru, prosjek brojeva u narančastom polju je 2.75, ovo je prosječno udruživanje s filtrom hiperparametra 2 koraka 2, može se odabrati i drugi hiperparametar.

Brojke koje izračunavaju i pohranjuju slojevi sažimanja mogu se zamisliti kao pokazatelji prisutnosti određene značajke. Kad bi se neuronska mreža oslanjala samo na izvornu mapu značajki, njezina bi sposobnost otkrivanja značajke ovisila o lokaciji na mapi. Na primjer, ako je broj 5 pronađen samo u gornjem lijevom kvadrantu, mreža bi naučila povezati značajku povezanu s brojem 5 s gornjim lijevom kvadrantom. Primjenom sažimanja izvlači se ova značajka na manju, općenitiju mapu koja samo pokazuje je li značajka prisutna u tom određenom kvadrantu ili ne. Sa svakim dodatnim slojem mapa se smanjuje čuvajući samo važne informacije o prisutnosti značajki od interesa. Kako mapa postaje manja, postaje sve neovisnija o lokaciji značajke. Sve dok je značajka otkrivena u približnoj blizini izvorne lokacije, trebala bi se na sličan način odražavati na mapi koju proizvode slojevi za udruživanje. Zbog svog fokusa na ekstremne vrijednosti, maksimalno udruživanje obraća pažnju na istaknutije značajke i rubove u receptivnom polju. Sažimanje prosjeka, s druge strane, stvara glađu mapu značajki jer rezultira prosjecima umjesto odabira ekstremnih vrijednosti. U praksi se maksimalno sažimanje primjenjuje češće jer je općenito bolje u identificiranju istaknutih značajki. U praktičnim primjenama, prosječno sažimanje koristi se samo za sažimanje mapa značajki na određenu veličinu. Zbog svoje sposobnosti sažimanja mapa značajki, udruživanje također može pomoći u klasifikaciji slika različitih veličina. Klasifikacijski sloj u neuronskoj mreži očekuje primanje ulaza u istom formatu. U skladu s tim, obično su slike u istoj standardnoj veličini. Variranjem odmaka tijekom operacije skupljanja, mogu se sažeti slike različitih veličina i još

uvijek proizvoditi mape značajki slične veličine. Općenito, sažimanje je posebno korisno pri zadatku klasifikacije slika gdje se treba otkriti prisutnost određenog objekta na slici, no nije bitno gdje se on točno nalazi. Činjenica da filtri za sažimanje koriste veći korak od konvolucijskih filtara i rezultiraju manjim izlazima također podupire učinkovitost mreže i dovodi do brže obuke. Drugim riječima, nepromjenjivost lokacije može uvelike poboljšati statističku učinkovitost mreže [36].

3.5.3. VGG16

VGG16 (*eng. Visual Geometry Group*) je vrsta CNN-a koja se smatra jednim od boljih modela računalnog vida do danas. Tvorci ovog modela procijenili su mreže i povećali dubinu korištenjem arhitekture s vrlo malim (3×3) konvolucijskim filtrima, što je pokazalo značajno poboljšanje u odnosu na prijašnje konfiguracije. Pomaknuli su dubinu na 16-19 težinskih slojeva, što čini približno 138 parametara koji se mogu trenirati. VGG16 je algoritam za otkrivanje i klasifikaciju objekata koji može klasificirati 1000 slika u 1000 različitih kategorija. To je jedan od popularnih algoritama za klasifikaciju slika i jednostavan je za korištenje s prijenosnim učenjem. Arhitektura VGG16 modela prikazana je na slici 16.



Slika 16. Arhitektura VGG16 modela [37]

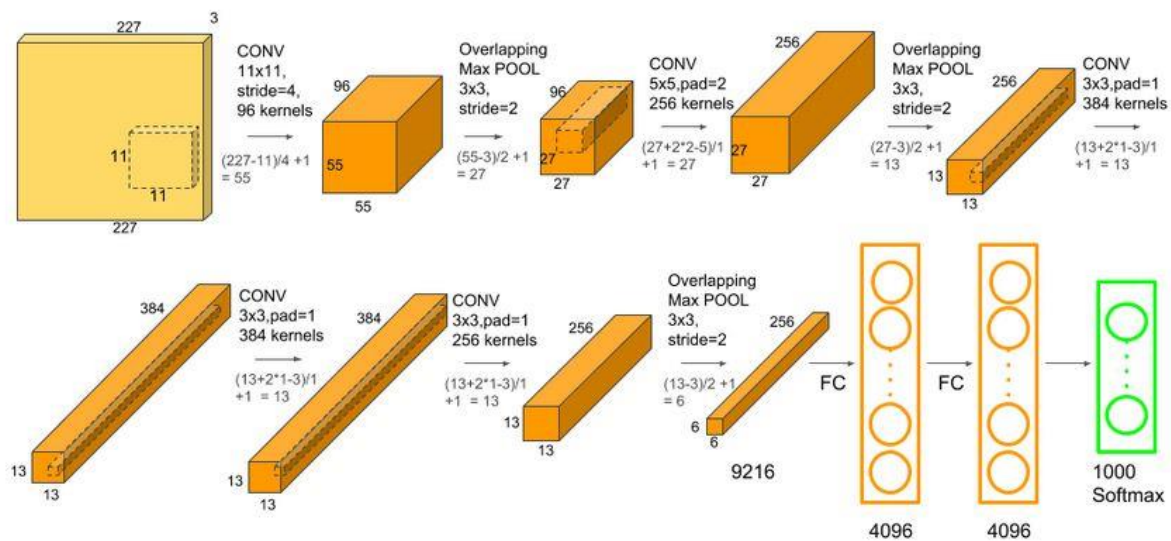
Značajke VGG16 modela [37]:

- Broj 16 u nazivu VGG16 odnosi se na 16 težinskih slojeva. U VGG16 postoji trinaest konvolucijskih slojeva, pet max-pooling slojeva i tri Dense sloja koji ukupno čine 21 sloj, ali ima samo šesnaest težinskih slojeva, tj. sloj parametara koji se mogu naučiti.
- VGG16 uzima veličinu ulaznog tenzora kao 224x224 s 3 RGB kanala.
- Najjedinstvenija stvar u vezi VGG16 je da su se umjesto velikog broja hiperparametara usredotočili na konvolucijske slojeve 3x3 filtra s korakom 1 i uvijek su koristili isti padding i maxpool sloj 2x2 filtra koraka 2.
- Slojevi konvolucije i max-poolinga dosljedno su raspoređeni kroz cijelu arhitekturu.
- Sloj Conv1 ima 64 filtera, Conv2 ima 128 filtera, Conv3 ima 256 filtera, Conv4 i Conv5 ima 512 filtera.

- Tri potpuno povezana (FC) sloja slijede niz konvolucijskih slojeva: prva dva imaju po 4096 neurona, treći izvodi 1000-smjernu ILSVRC klasifikaciju i stoga sadrži 1000 neurona (po jedan za svaku klasu). Izlazni sloj se sastoji od softmax aktivacijske funkcije.

3.5.4. AlexNet

AlexNet je naziv konvolucijske neuronske mreže koja je imala veliki utjecaj na polje strojnog učenja, posebno u primjeni dubokog učenja. Predložio ga je Alex Krizhevsky za ImageNet Large Scale Visual Recognition Challenge (ILSVRC) gdje je pobijedio na natjecanju 2012.godine. ILSVRC je natjecanje u kojem istraživački timovi ocjenjuju svoje algoritme na ogromnom skupu podataka označenih slika (ImageNet) i natječu se za postizanje veće točnosti na nekoliko zadataka vizualnog prepoznavanja. Mreža je imala vrlo sličnu arhitekturu kao LeNet, ali je bila dublja i s više filtara po sloju i poredanim konvolucijskim slojevima. Arhitektura je u top 5 sa stopom pogreške od 15.3%, dok je sljedeći najbolji rezultat ostao je daleko iza s 26.2%. AlexNet arhitektura prikazana na slici 17. se sastoji od ukupno osam slojeva, od kojih su prvih 5 konvolucijski slojevi, a posljednja 3 su potpuno povezana. Prva dva konvolucijska sloja povezana su s preklapajućim slojevima maksimalnog skupljanja kako bi se izdvojio maksimalan broj značajki. Treći, četvrti i peti konvolucijski slojevi izravno su povezani s potpuno povezanim slojevima. Svi izlazi konvolucijskih i potpuno povezanih slojeva povezani su s ReLU nelinearnom aktivacijskom funkcijom. Konačni izlazni sloj povezan je sa softmax aktivacijskim slojem, koji proizvodi distribuciju od 1000 oznaka klasa [38].



Slika 17. Arhitektura AlexNet modela [38]

Značajke AlexNet modela [38]:

- AlexNet je bila prva konvolucijska mreža koja je koristila GPU za povećanje performansi.
- AlexNet arhitektura sastoji se od 5 konvolucijskih slojeva, 3 max-pooling sloja, 2 normalizacijska sloja, 2 potpuno povezana sloja i 1 softmax sloja.
- Svaki konvolucijski sloj sastoji se od konvolucijskih filtara i nelinearne aktivacijske funkcije ReLU.
- Slojevi sažimanja koriste se za izvođenje maksimalnog udruživanja.
- Veličina ulaza je fiksna zbog prisutnosti potpuno povezanih slojeva.
- AlexNet ima veličinu ulaznog tenzora kao 227x 227 s 3 RGB kanala.

3.5.5. ResNet50

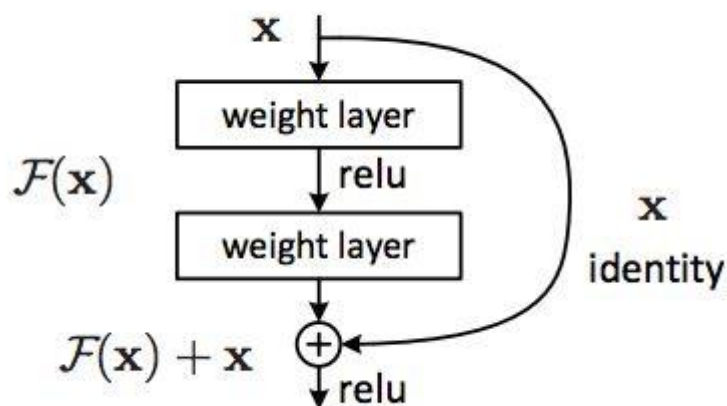
ResNet (*eng. Residual Networks*) inovativna je neuronska mreža koja se koristi kao oslonac za mnoge zadatke računalnog vida, a temeljni napredak je što omogućuje treniranje iznimno dubokih neuronskih mreža s više od 150 slojeva. To je inovativna neuronska mreža koju su prvi predstavili Kaiming He, Xiangyu Zhang, Shaoqing Ren i Jian Sun u svom istraživačkom radu o računalnom vidu iz 2015.godine pod nazivom „Deep Residual Learning for Image Recognition“ [39]. ResNet ima mnogo varijanti koje rade na istom konceptu, ali imaju različiti

broj slojeva. ResNet50 se koristi za označavanje varijante koja može raditi s 50 slojeva neuronske mreže.

ResNet ima veliku važnost u računalnom vidu. Gradnjom duboke konvolucijske neuronske mreže uspješno se postići visoke vrijednosti točnosti, no dublje mreže predstavljaju i složenije značajke što dovodi do povećanja robusnosti modela i performansi. Dodatni slojevi pomažu u učinkovitijem rješavanju složenih problema jer se različiti slojevi mogu osposobiti za različite zadatke kako bi se dobili vrlo precizni rezultati. Dok broj poredanih slojeva može obogatiti značajke modela, dublja mreža može pokazati problem degradacije. Drugim riječima, kako se broj slojeva neuronske mreže povećava, razine točnosti mogu postati zasićene. Rezultat toga je pogoršanje izvedbe modela na podacima za treniranje i testiranje. Degradacija nije rezultat prekomjernog overfittinga, već umjesto toga može proizaći iz inicijalizacije mreže, funkcije optimizacije ili, još važnije, problema nestajanja ili eksploziranja gradijenata. ResNet je nastao s ciljem rješavanja upravo ovog problema. Duboke rezidualne mreže koriste rezidualne blokove za poboljšanje točnosti modela. Koncept „skip connections“, koji leži u srži rezidualnih blokova, snaga je ove vrste neuronske mreže. Skip connections radi na dva načina:

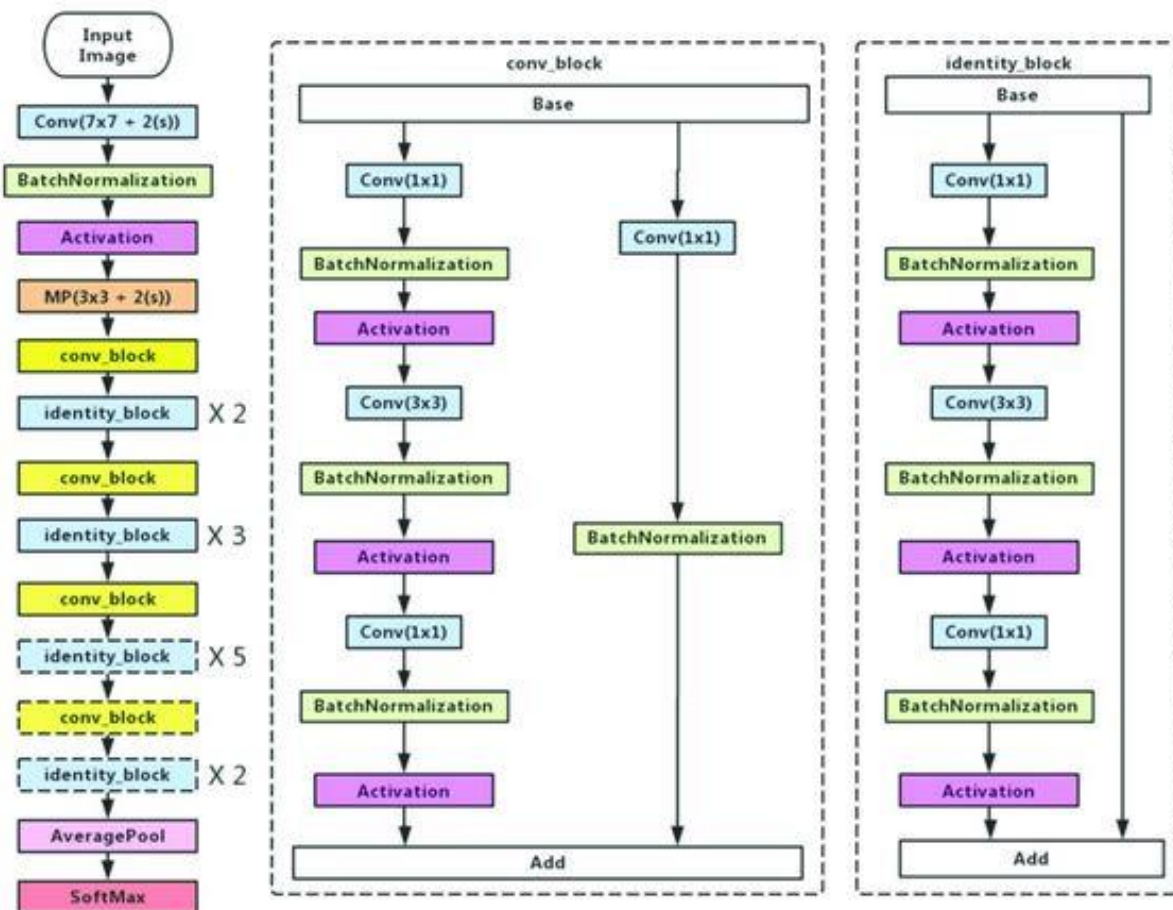
- Ublažava problem nestajanja gradijenta postavljanjem alternativnog prečaca kroz koji gradijent prolazi.
- Omogućuju modelu da nauči funkciju identiteta te time osigurava da viši slojevi modela ne rade ništa lošije od nižih slojeva.

Rezidualni blokovi prikazani na slici 18. znatno olakšavaju slojevima učenje funkcija identiteta. Kao rezultat toga, ResNet poboljšava učinkovitost dubokih neuronskih mreža s više neuronskih slojeva dok minimizira pogrešku. Drugim riječima, veze preskakanja dodaju izlaze s prethodnih slojeva na izlaze poredanih slojeva, što omogućuje treniranje mnogo dubljih mreža nego što je to bilo moguće prije.



Slika 18. Residual blok [40]

Arhitektura ResNet50 ima 4 faze kao što je prikazano na slici 19. Mreža može uzeti ulaznu sliku koja ima visinu i širinu kao višekratnike 32 i 3 kao širinu kanala. Svaka ResNet arhitektura izvodi početnu konvoluciju i max-pooling koristeći 7×7 odnosno 3×3 veličine kernela. Nakon toga počinje 1. faza mreže i ima 3 rezidualna bloka od kojih svaki sadrži 3 sloja. Veličina kernela koja se koristi za izvođenje operacije konvolucije u sva 3 sloja bloka stupnja 1 je 64, 64 i 128 redom. Pune strelice odnose se na vezu identiteta, a isprekidana povezana strelica predstavlja da se operacija konvolucije u zaostalom bloku izvodi s korakom 2. Zbog toga će se veličina unosa smanjiti na pola u smislu visine i širine, ali će se širina kanala udvostručiti. Širina kanala se udvostručuje kako mreža napreduje iz jedne faze u drugu, a veličina ulaza smanjuje se na pola. Koristi se dizajn uskog grla za duboku mrežu kao što je ResNet50. Svaka funkcija reziduala ima 3 sloja poredanih jedan preko drugog. Tri konvolucijska sloja su 1×1 , 3×3 , 1×1 , gdje su konvolucijski slojevi 1×1 odgovorni za smanjivanje i zatim vraćanje dimenzija, a sloj 3×3 ostavljen je kao usko grlo s manjim ulazno/izlaznim dimenzijama. Na samom kraju mreža ima sloj prosječnog skupljanja praćen potpuno povezanim slojem koji ima 1000 neurona [39].



Slika 19. Arhitektura ResNet50 [41]

3.5.6. DenseNet121

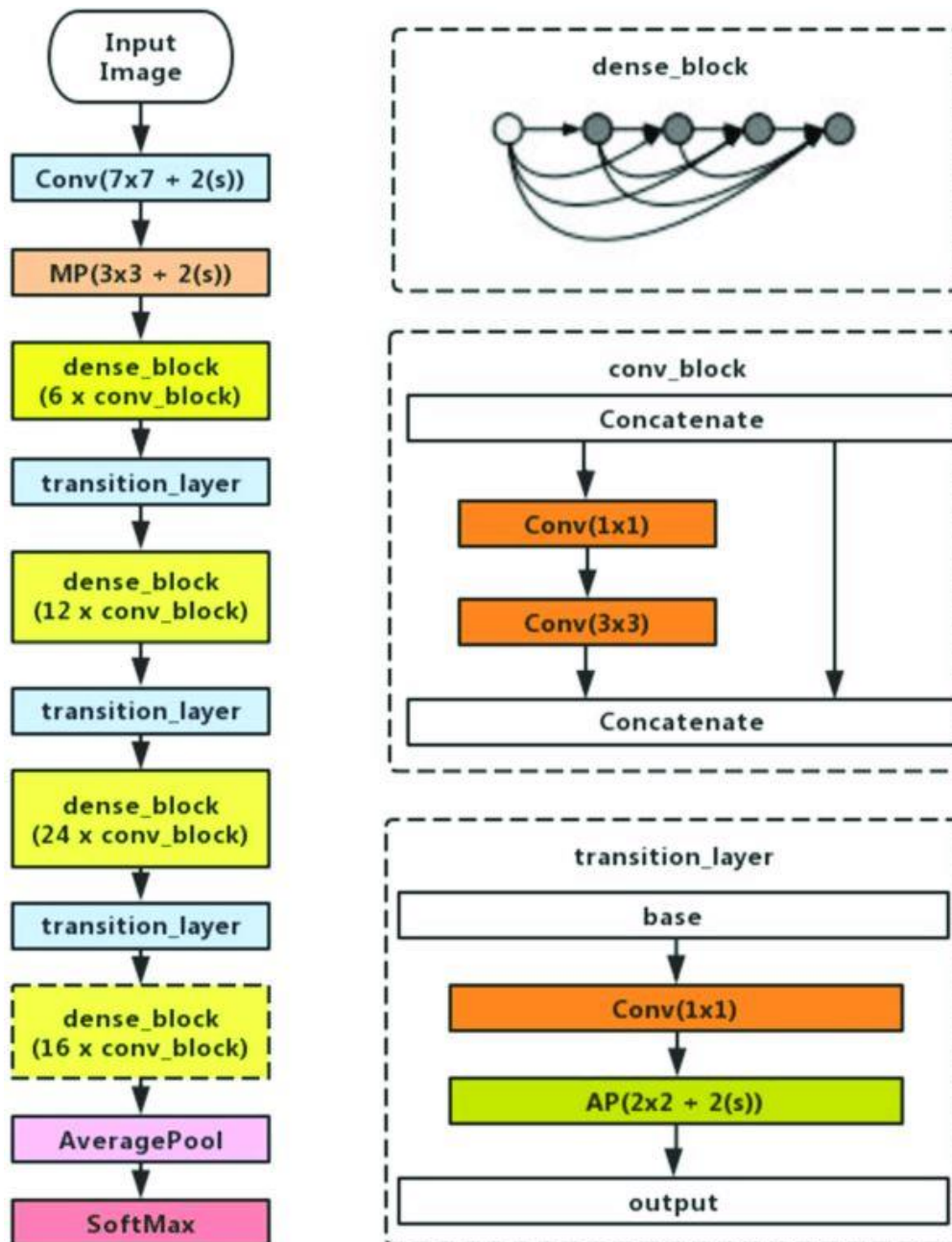
DenseNet (*eng. Densely Connected Convolutional Networks*) je vrsta konvolucijske neuronske mreže te je jedno od novih otkrića za vizualno prepoznavanje objekata. DenseNet je poprilično sličan ResNetu s nekim temeljnim razlikama. ResNet koristi aditivnu metodu koja spaja prethodni sloj (identitet) s budućim slojem, dok DenseNet spaja izlaz prethodnog sloja s budućim slojem, koristeći veze između slojeva kroz guste blokove gdje povezuje sve slojeve izravno jedan s drugim. Kako bi se očuvala priroda povratne informacije, svaki sloj dobiva dodatne ulaze od svih prethodnih slojeva i prosleđuje vlastite mape značajki svim sljedećim slojevima. Povezivanjem na ovaj način DenseNet zahtjeva manje parametre od ekvivalentnog tradicionalnog CNN-a, budući da nema potrebe za učenjem mapa suvišnih značajki. DenseNet je razvijen posebno za poboljšanje smanjene točnosti uzrokovane nestajanjem gradijenta u

neuronskim mrežama. Jednostavnije rečeno, zbog duljeg puta između ulaznog sloja i izlaznog sloja, informacija nestaje prije nego što stigne na odredište [42].

Postoje tri vrste blokova u DenseNet implementaciji:

1. Konvolucijski blok (*eng. Convolution block*) - osnovni blok gustog bloka, sličan je bloku identiteta u ResNetu.
2. Gusti blok (*eng. Dense block*) - glavna komponenta u DenseNetu. U njemu su konvolucijski blokovi ulančani i gusto povezani.
3. Prijelazni sloj (*eng. Transition layer*) - povezuje dva susjedna gusta bloka. Budući da su veličine mape značajki iste unutar gustog bloka, prijelazni sloj smanjuje dimenzije mape značajki. Tehnika dizajna uskog grla usvojena je u svim blokovima.

Arhitektura DenseNet121 prikazana je na slici 20. Svaki gusti blok ima različit broj slojeva (ponavljanja) od kojih svaki sadrži dvije konvolucije: kernel veličine 1x1 kao sloj uskog grla i kernel 3x3 za izvođenje operacije konvolucije. Svaki prijelazni sloj ima 1x1 konvolucijski sloj i 2x2 sloj prosječnog udruživanja s korakom od 2. Ukratko DenseNet121 sadrži 120 konvolucijskih slojeva i 4 average poolinga. Svi slojevi koji se nalaze unutar istog gustog bloka i prijelaznih slojeva raspoređuju svoje težine na višestruke ulaze što omogućuje dubljim slojevima da koriste rano izdvojene značajke. Budući da prijelazni slojevi izlaze s mnogo suvišnih značajki, slojevi u drugom i trećem gustom bloku dodjeljuju najmanje težine izlazu prijelaznih slojeva. Bez obzira što se težine cijelog gustog bloka koriste u konačnim slojevima, još uvijek može postojati više značajki visoke razine generiranih dublje u modelu [42].



Slika 20. Arhitektura DenseNet121 [41]

3.5.7. Prilagođena arhitektura

Pri stvaranju prilagođene arhitekture, to jest ručno prilagođenog modela, korišteno je teoretsko znanje opisano u prijašnjim dijelovima rada. Na slici 21. prikazana je prilagođena arhitektura modela. U početku je dodan jedan konvolucijski sloj od 16 filtera veličine filtera 3 x 3. Razlog za postavljanje malog broja filtera je otkrivanje rubova, kutova i linija. Nakon toga je dodan sloj sažimanja s 2 x 2 filtrom kako bi se izlučile značajke te slike, te je zatim broj konvolucijskih

slojeva i broj filtara povećan na 32, 64 i 128 koji imaju isti 3 x 3 karnel. To kombinira ove male uzorke kako se broj filtara povećava i pronalazi veće uzorke poput kvadrata, krugova itd. Na njih su primijenjeni slojevi maksimalnog sažimanja kako bi se dobio maksimum. Na kraju je primijenjen potpuno povezani sloj od 256 neurona zajedno s softmax aktivacijskom funkcijom koji izračunava ocjenu vjerojatnosti za svaku klasu i klasificira oznake konačne odluke. Primijenjena je aktivacijska funkcija ReLU u svakom konvolucijskom sloju. Aktivacijska funkcija pretvara ulazni ponderirani zbroj u izlaz tog čvora. Primjer prilagođene arhitekture prikazana je na slici 22.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 222, 222, 16)       160
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)       0
conv2d_1 (Conv2D)            (None, 109, 109, 32)       4640
conv2d_2 (Conv2D)            (None, 107, 107, 32)       9248
max_pooling2d_1 (MaxPooling2D) (None, 53, 53, 32)         0
conv2d_3 (Conv2D)            (None, 51, 51, 64)         18496
conv2d_4 (Conv2D)            (None, 49, 49, 64)         36928
conv2d_5 (Conv2D)            (None, 47, 47, 64)         36928
max_pooling2d_2 (MaxPooling2D) (None, 23, 23, 64)         0
conv2d_6 (Conv2D)            (None, 23, 23, 128)        73856
conv2d_7 (Conv2D)            (None, 23, 23, 128)        147584
max_pooling2d_3 (MaxPooling2D) (None, 11, 11, 128)        0
flatten (Flatten)            (None, 15488)              0
dense (Dense)                 (None, 256)                3965184
dense_1 (Dense)               (None, 4)                  1028
-----
Total params: 4,294,052
Trainable params: 4,294,052
Non-trainable params: 0

```

Slika 21. Prilagođena arhitektura

3.6. Evaluacijske metrike

Evaluacijske metrike koriste se za mjerenje kvalitete statističkog modela ili modela strojnog učenja. Postoji mnogo različitih vrsta mjernih podataka za procjenu kao što je točnost klasifikacije, logaritamski gubitak, matrica konfuzije i druge. Vrlo je važno koristiti evaluacijske metrike za procjenu modela, zato što model može imati dobre rezultate korištenjem jednog mjerenja iz jedne evaluacijske metrike, ali može imati loše rezultate koristeći drugo mjerenje iz druge evaluacijske metrike. Korištenje evaluacijske metrike ključno je za osiguravanje da model radi ispravno i optimalno.

Točnost (*eng. Accuracy*) je najjednostavnija metrika za procjenu modela. Predstavlja omjer točnih i ukupnog broja predviđanja napravljenih za skup podataka te je izražena formulom (3.11):

$$\text{Točnost} = \frac{\text{točne predikcije}}{\text{ukupan broj predikcija}} \cdot \quad (3.11.)$$

Točnost je korisna kada je ciljna klasa dobro uravnotežena, ali nije dobar izbor s neuravnoteženim klasama. Točnost daje cjelokupnu sliku o tome koliko se može osloniti na predviđanje modela. Ova metrika ne vidi razliku između klasa i vrsta pogrešaka zato nije dovoljno dobra za neuravnotežene skupove podataka.

U nastavku su navedene 4 metrike često korištene za problem klasifikacije:

- Točnost koja se može izračunati u smislu pozitivnih i negativnih vrijednosti za binarnu klasifikaciju, a predstavlja postotak svih točno predviđenih vrijednosti u odnosu na sve vrijednosti. Matematički opis dan je izrazom (3.12.):

$$\text{Točnost} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.12.)$$

gdje je:

TP (*eng. True positive*) – Broj opservacija koje su ispravno klasificirane kao pozitivne.

TN (*eng. False positive*) – Broj opservacija koje su klasificirane kao pozitivne, ali su zapravo negativne.

FP (*eng. True negative*) – Broj opservacija koje su ispravno klasificirane kao negativne.

FN (*eng. False negative*) – Broj opservacija koje su klasificirane kao negativne, ali su zapravo pozitivne.

- Preciznost (*eng. Precision*) predstavlja omjer stvarno pozitivnih rezultata prema svim pozitivnim rezultatima koje predviđa model. Korisno je za iskrivljene i neuravnotežene skupove podataka. Što više lažno pozitivnih rezultata model predviđa, to je manja preciznost. Matematički opis dan je izrazom (3.13.):

$$\text{Preciznost} = \frac{TP}{TP + FP} . \quad (3.13.)$$

- Odaziv (*eng. Recall*) predstavlja omjer stvarno pozitivnih rezultata u odnosu na sve pozitivne rezultate u skupu podataka. Mjeri sposobnost modela da otkrije pozitivne uzorke. Što više lažno negativnih rezultata model predvidi, to je manja vrijednost metrike. Matematički opis dan je izrazom (3.14.):

$$\text{Odaziv} = \frac{TP}{TP + FN} \quad (3.14.)$$

- F1-score je metrika koja kombinira i preciznost i odaziv. Što je viši rezultat F1, to je bolja izvedba modela. Raspon za F1 rezultat je [0,1]. Matematički opis dan je izrazom (3.15.):

$$F1 = 2 \frac{\text{preciznost} * \text{odaziv}}{\text{preciznost} + \text{odaziv}} \quad (3.15.)$$

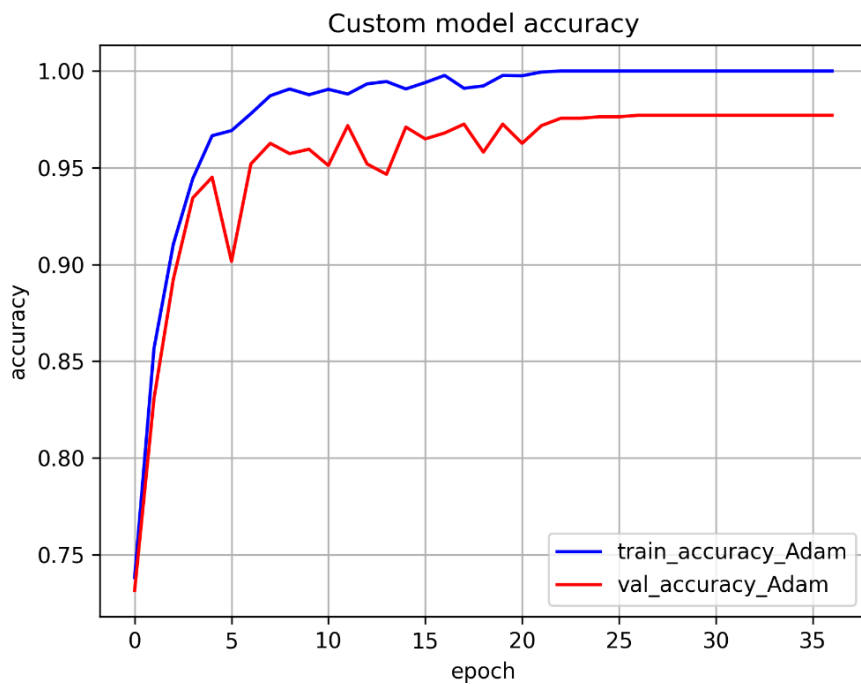
4. REZULTATI

Provedene su eksperimentalne procjene kako bi se utvrdio značaj i točnost prilagođene arhitekture u kojoj su trenirani svi slojevi. Model je konfiguriran sa inicijalnom stopom učenja od 0.001 i epohama 100 za sva tri optimizacijska algoritma koji se ispituju. Kao funkcija gubitaka za potrebe ovog istraživanja korištena je unakrsna entropija. Većina tehnika koje se temelje na neuronskim mrežama, uključujući CNN, koristi pad gradijenta za smanjenje stope pogreške u procesu treniranja i za ažuriranje unutarnjih parametara. Gradijentni spust je optimizacijski algoritam prvog reda i njegova derivacija daje smjer i povećava ili smanjuje funkciju pogreške. Kao tehnika zaštite od overfittinga korištena je tzv. early stopping funkcija koja prekida treniranje ukoliko se vrijednost točnosti na validacijskom podacima nije poboljšala u proteklih 10 epoha. U slučaju prilagođene arhitekture, treniranje je prekinuto na sljedećem broju epoha: Adam – 26, SGD – 63 i RMSprop – 16. U tablici 3. prikazane su izračunate vrijednosti evaluacijskih metrika za svaki optimizacijski algoritam.

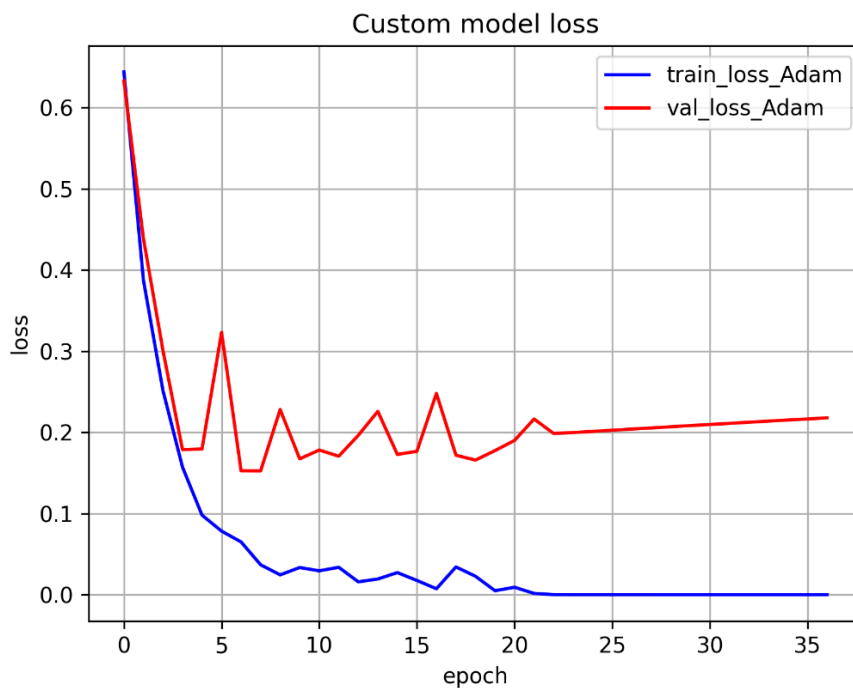
Tablica 3. Performanse prilagođene arhitekture za različite optimizacijske algoritme

Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1	Overfitting
Adam	0.9771	0.9759	0.9753	0.9756	Da
SGD	0.9634	0.9614	0.9605	0.9608	Da
RMSprop	0.9741	0.9729	0.9721	0.9721	Da

Prema tablici 3. moguće je vidjeti kako se Adam optimizacijski algoritam pokazao najbolji s rezultatom točnosti od 97.7% u 26 epohi. Na slikama 22. i 23. prikazana je točnost i gubitak prilagođenog modela na trening i validacijskom skupu podataka korištenjem Adam optimizacijskog algoritma, a ostale slike grafova za SGD i RMSprop optimizacijske algoritme nalaze se u poglavlju Dodaci.



Slika 22. Točnost prilagođenog modela koristeći Adam optimizacijski algoritam



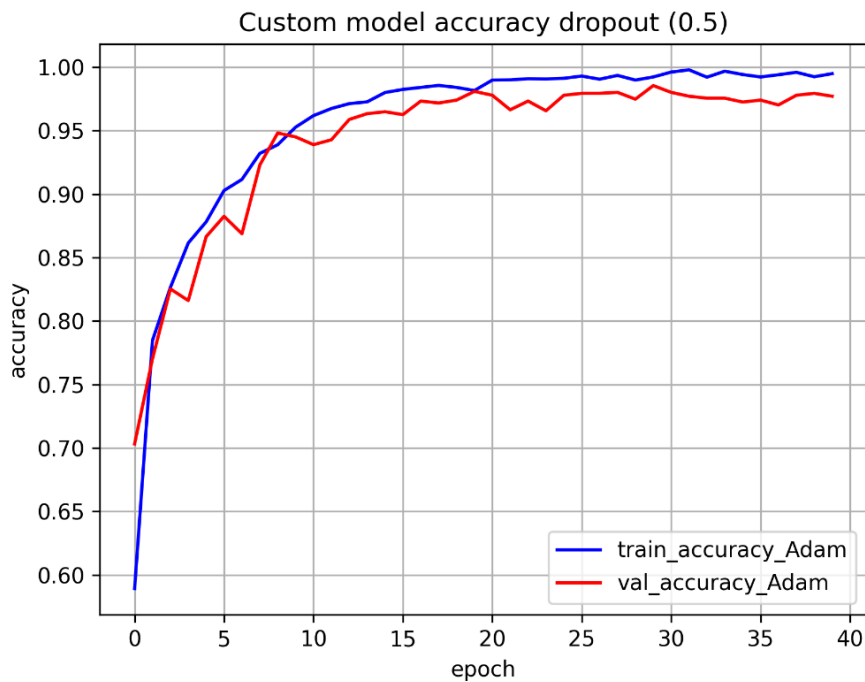
Slika 23. Gubitak prilagođenog modela koristeći Adam optimizacijski algoritam

Kako bi se reducirao overfitting dodaje se dropout(0.5) prije izlaznog sloja. U tablici 4. su prikazani novi najbolji rezultati u 29 epohi s izmijenjenom stopom učenja od 0.0001.

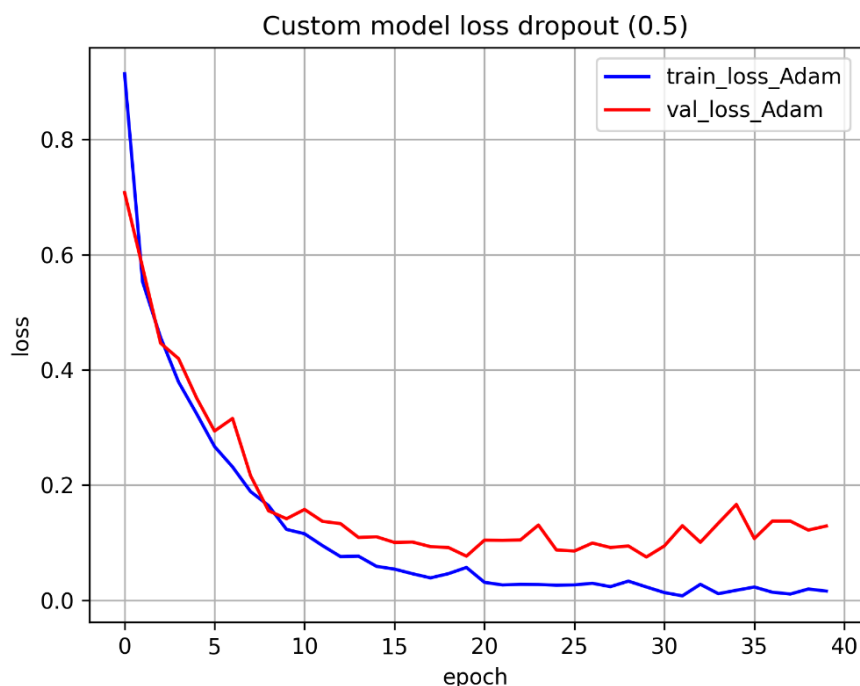
Tablica 4. Rezultati nakon dodavanja dropota(0.5)

Optimizacijski algoritam	Točnost	Preciznost	Odaziv	F1
Adam	0.9855072464	0.9851275624	0.9843137255	0.9847022647

Na slikama 24. i 25. prikazana je točnost i gubitak prilagođenog modela na trening i validacijskom skupu podataka s modificiranom arhitekturom i stopom učenja.



Slika 24. Točnost prilagođenog modela koristeći dropout(0.5) kod Adam optimizacijskog algoritam



Slika 25. Gubitak prilagođenog modela koristeći dropout(0.5) kod Adam optimizacijskog algoritma

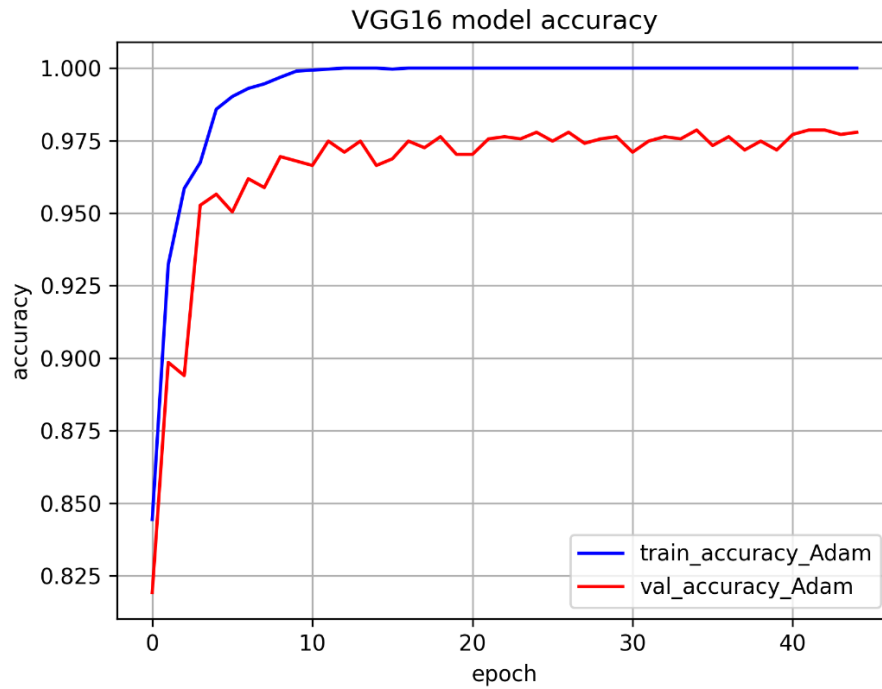
U VGG16 arhitekturi treniran je samo zadnji sloj s obzirom da je korišten predtrenirani model na ImageNet skupu podataka. Model je konfiguriran s bitnim hiperparametrima kao što je optimizacijski algoritam Adam, SGD i RMSprop te sa stopom učenja od 0.001 i epohama 100. U slučaju VGG16 arhitekture, treniranje je prekinuto na sljedećem broju epoha: Adam – 34, SGD – 73 i RMSprop – 26. U tablici 5. prikazane su izračunate vrijednosti evaluacijskih metrika za svaki optimizacijski algoritam.

Tablica 5. Performanse VGG16 arhitekture za različite optimizacijske algoritme

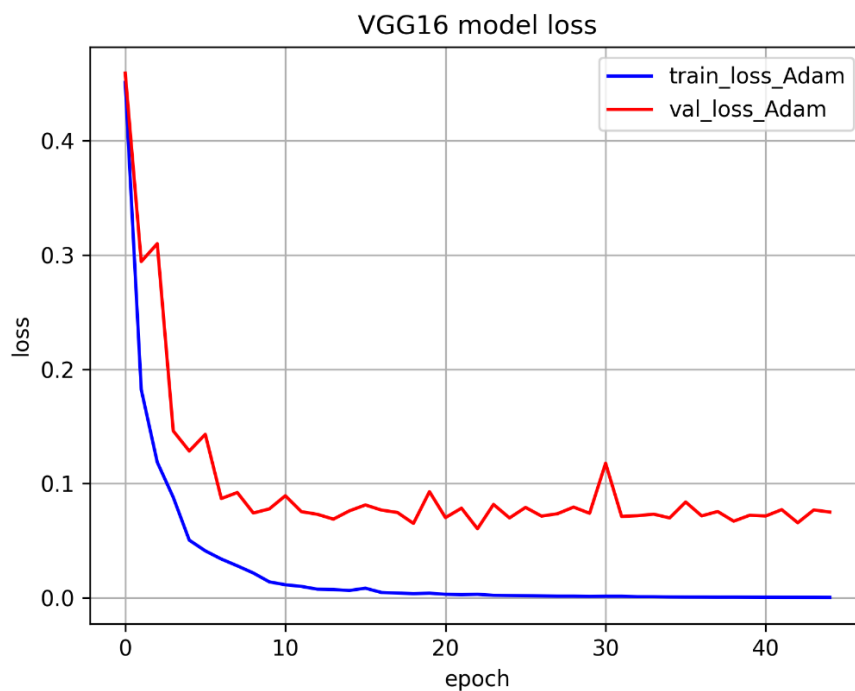
Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1	Overfitting
Adam	0.9786	0.9780	0.9768	0.9770	Da
SGD	0.9336	0.9328	0.9301	0.9307	Da
RMSprop	0.9756	0.9751	0.9734	0.9738	Da

Prema tablici 5. moguće je vidjeti kako se Adam optimizacijski algoritam pokazao najbolji s rezultatom točnosti od 97.9% u 34 epohi. Na slikama 26. i 27. prikazana je točnost i gubitak

VGG16 modela na trening i validacijskom skupu podataka korištenjem Adam optimizacijskog algoritma, a ostale slike grafova za SGD i RMSprop optimizacijske algoritme prikazane su pod poglavljem Dodaci.



Slika 26. Točnost VGG16 modela koristeći Adam optimizacijski algoritam



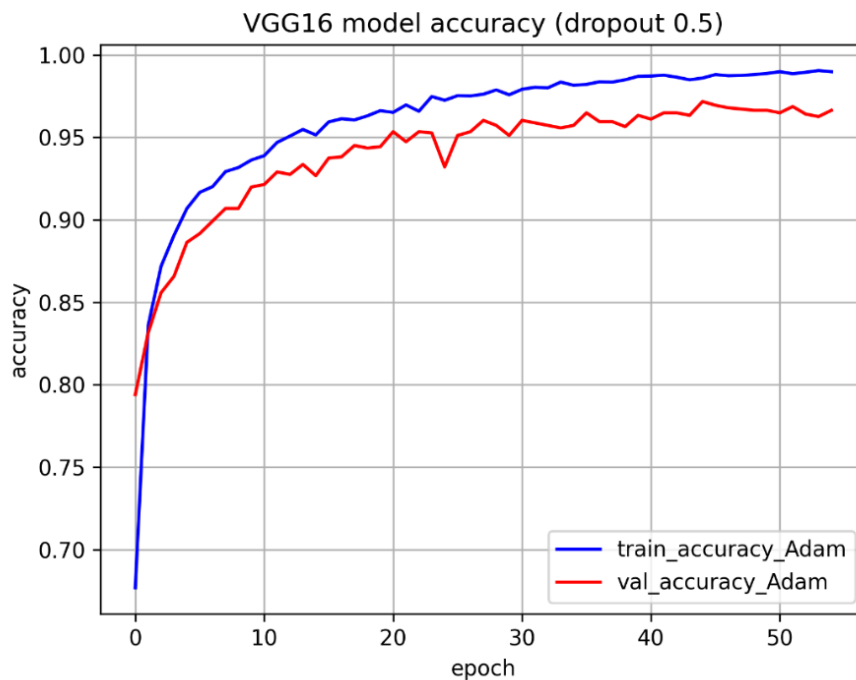
Slika 27. Gubitak VGG16 modela koristeći Adam optimizacijski algoritam

Kako bi se reducirao overfitting dodaje se dropout(0.5) prije izlaznog sloja. U tablici 6. su prikazani novi najbolji rezultati u 44 epohi s izmijenjenom stopom učenja 0.0001.

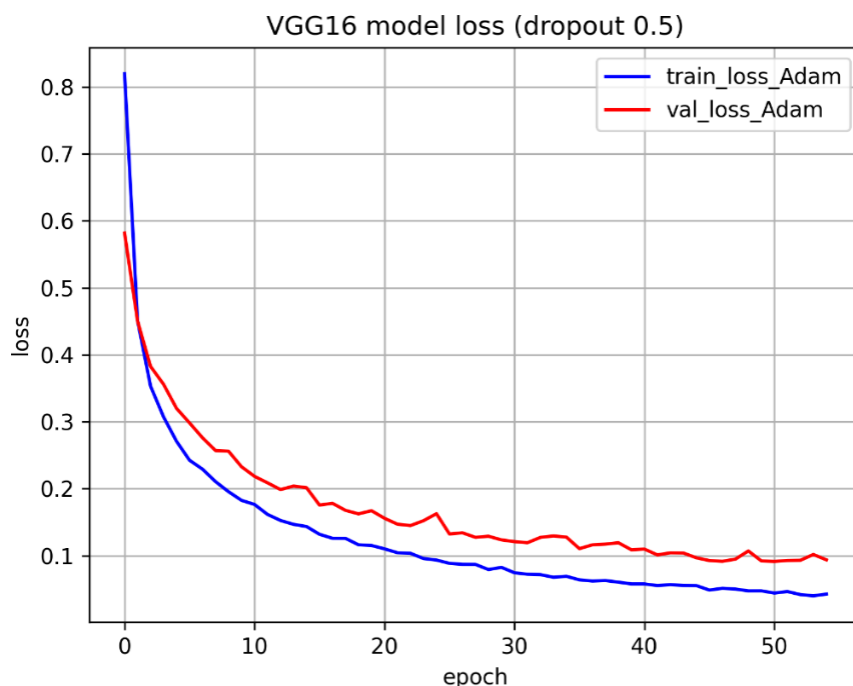
Tablica 6. Rezultati nakon dodavanja dropouta(0.5)

Optimizacijski algoritam	Točnost	Preciznost	Odaziv	F1
Adam	0.9717772693	0.9702140663	0.9693627451	0.9696151818

Na slikama 28. i 29. prikazana je točnost i gubitak VGG16 modela na trening i validacijskom skupu podataka dodavanjem dropouta(0.5) za reduciranje overfittinga Adam optimizacijskog algoritma.



Slika 28. Točnost VGG16 modela koristeći dropout(0.5) kod Adam optimizacijskog algoritam



Slika 29. Gubitak VGG16 modela koristeći dropout(0.5) kod Adam optimizacijskog algoritma

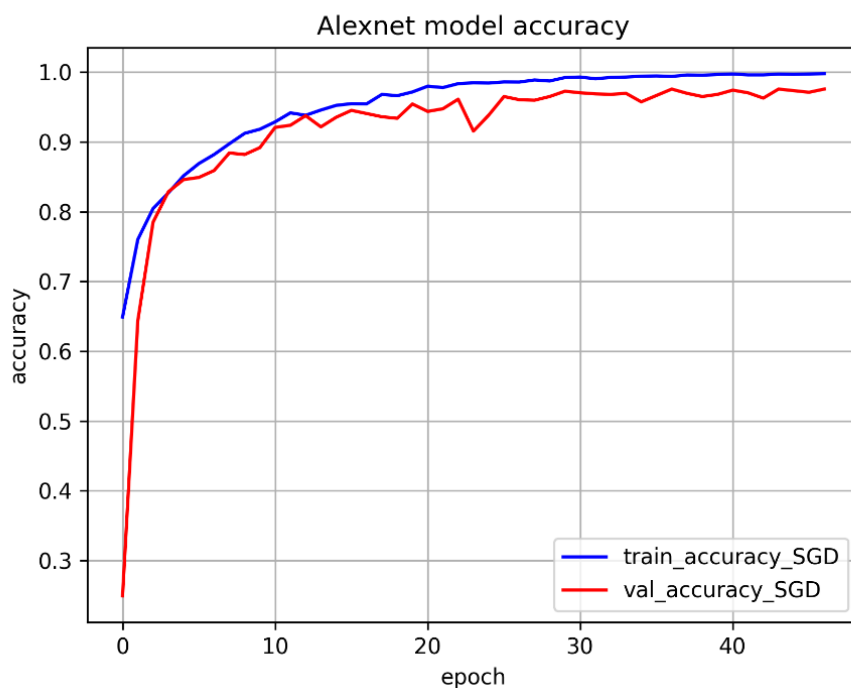
U AlexNet arhitekturi treniraju se svih 8 slojeva od koliko se arhitektura i sastoji, od kojih su prvih 5 konvolucijski slojevi, a posljednja 3 su potpuno povezana. Model je konfiguriran s bitnim hiperparametrima kao što je optimizacijski algoritam Adam, SGD i RMSprop te sa stopom učenja od 0.001 i epochama 100. U slučaju AlexNet arhitekture, treniranje je prekinuto na sljedećem broju epoaha: Adam – 35, SGD –36 i RMSprop – 36.

U tablici 7. prikazane su izračunate vrijednosti evaluacijskih metrika za svaki optimizacijski algoritam.

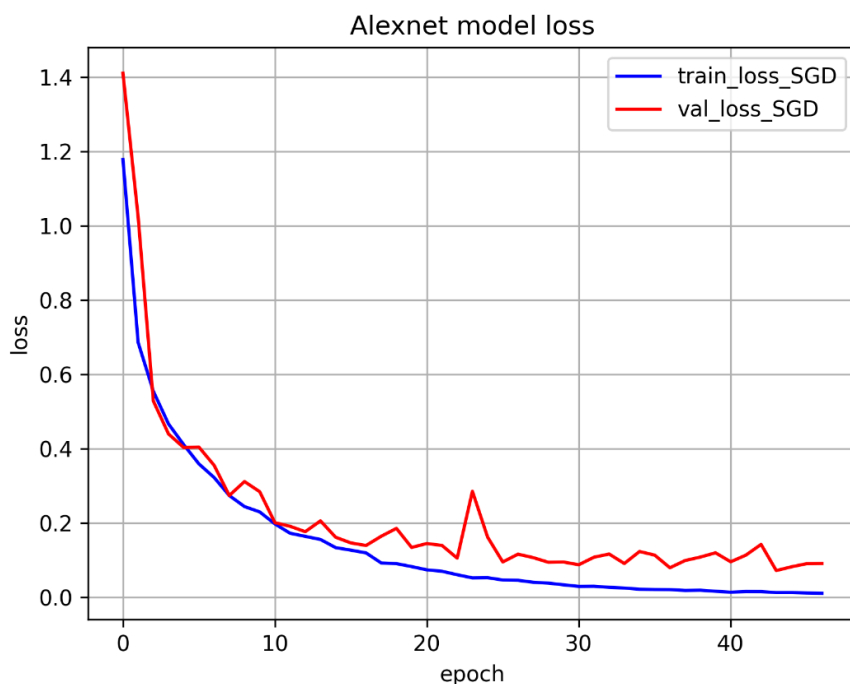
Tablica 7. Performanse AlexNet arhitekture za različite optimizacijske algoritme

Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1	Overfitting
Adam	0.9580	0.9577	0.9546	0.9558	Ne
SGD	0.9756	0.9738	0.9736	0.9737	Ne
RMSprop	0.9504	0.9500	0.9468	0.9481	Ne

Prema tablici 5. moguće je vidjeti kako se SGD optimizacijski algoritam pokazao najbolji s rezultatom točnosti od 97.6% u 36 epohi. Na slikama 30. i 31. prikazana je točnost i gubitak AlexNet modela na trening i validacijskom skupu podataka korištenjem SGD optimizacijskog algoritma, a ostale slike grafova za Adam i RMSprop optimizacijske algoritme prikazani su pod poglavljem Dodaci. S obzirom da nije došlo do overfittinga, nisu potrebne dodatne modifikacije arhitekture i hiperparametara.



Slika 30. Točnost AlexNet modela koristeći SGD optimizacijski algoritam



Slika 31. Gubitak AlexNet modela koristeći SGD optimizacijski algoritam

U ResNet50 arhitekturi treniran je samo zadnji sloj s obzirom da je korišten predtrenirani model na ImageNet skupu podataka. Model je konfiguriran s bitnim hiperparametrima kao što je optimizacijski algoritam Adam, SGD i RMSprop te sa stopom učenja od 0.001 i epohama 100. U slučaju ResNet50 arhitekture, treniranje je prekinuto na sljedećem broju epoha: Adam – 61, SGD – 30 i RMSprop – 95.

U tablici 8. prikazane su izračunate vrijednosti evaluacijskih metrika za svaki optimizacijski algoritam.

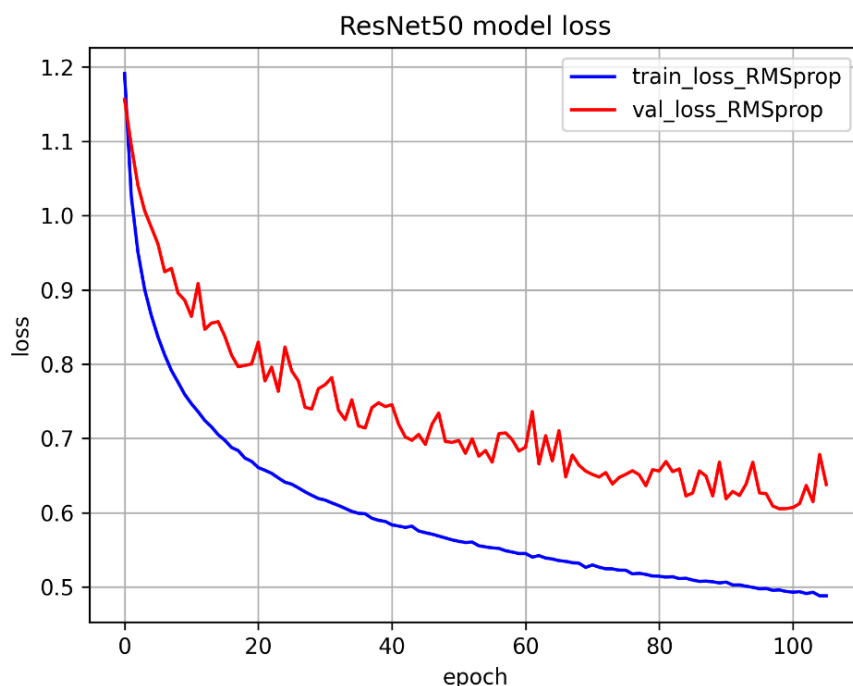
Tablica 8. Performanse ResNet50 arhitekture za različite optimizacijske algoritme

Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1	Overfitting
Adam	0.7437	0.7269	0.7314	0.7241	Da
SGD	0.5767	0.5636	0.5676	0.5509	Da
RMSprop	0.7536	0.7349	0.7420	0.7300	Da

Prema tablici 8. moguće je vidjeti kako se RMSprop optimizacijski algoritam pokazao najbolji s rezultatom točnosti od 75.4% u 95 epohi. Na slikama 32. i 33. prikazana je točnost i gubitak ResNet50 modela na trening i validacijskom skupu podataka korištenjem RMSprop optimizacijskog algoritma, a ostale slike grafova za Adam i SGD optimizacijske algoritme prikazani su pod poglavljem Dodaci.



Slika 32. Točnost ResNet50 modela koristeći RMSprop optimizacijski algoritam



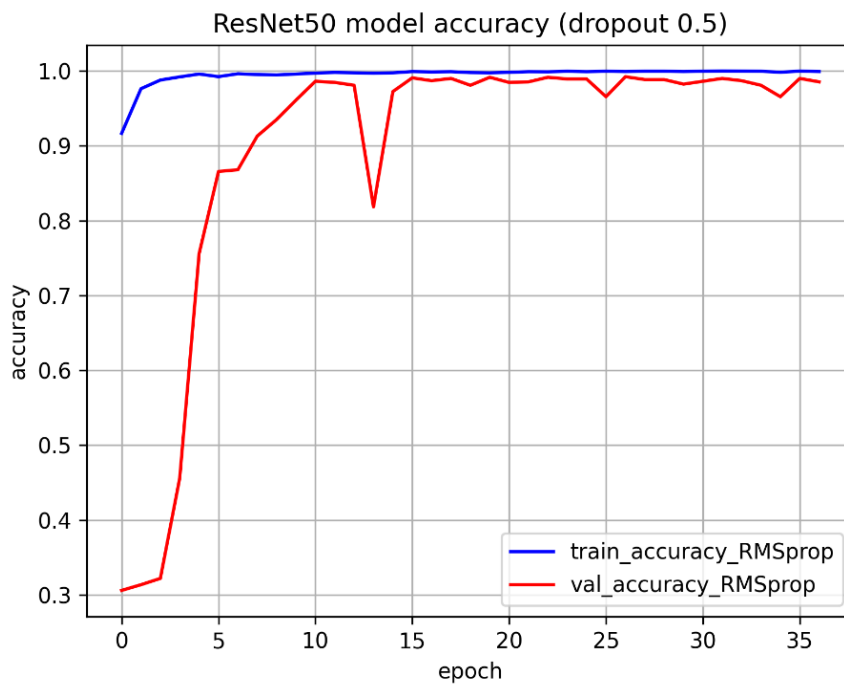
Slika 33. Gubitak ResNet50 modela koristeći RMSprop optimizacijski algoritam

Kako bi se reducirao overfitting dodaje se dropout(0.5) prije izlaznog sloja. U tablici 9. su prikazani novi najbolji rezultati u 26 epohi s izmijenjenom stopom učenja 0.0001.

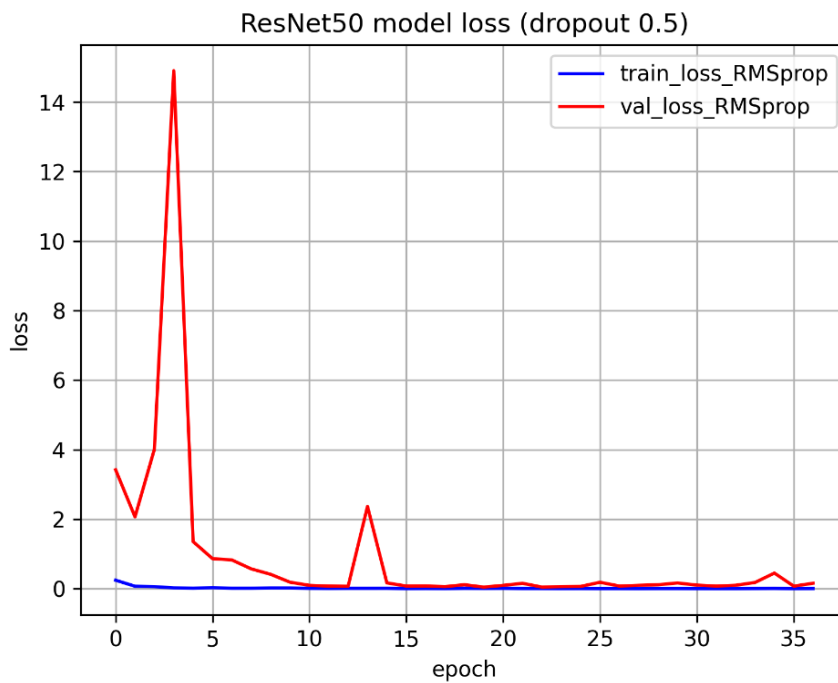
Tablica 9. Rezultati nakon dodavanja dropouta(0.5)

Optimizacijski algoritam	Točnost	Preciznost	Odaziv	F1
RMSprop	0.9923722349	0.9917624031	0.9917973856	0.9917273002

Na slikama 34. i 35. prikazana je točnost i gubitak ResNet50 modela na trening i validacijskom skupu podataka dodavanjem dropouta(0.5) za reduciranje overfittinga RMSprop optimizacijskog algoritma.



Slika 34. Točnost ResNet50 modela koristeći dropout(0.5) kod RMSprop optimizacijskog algoritma



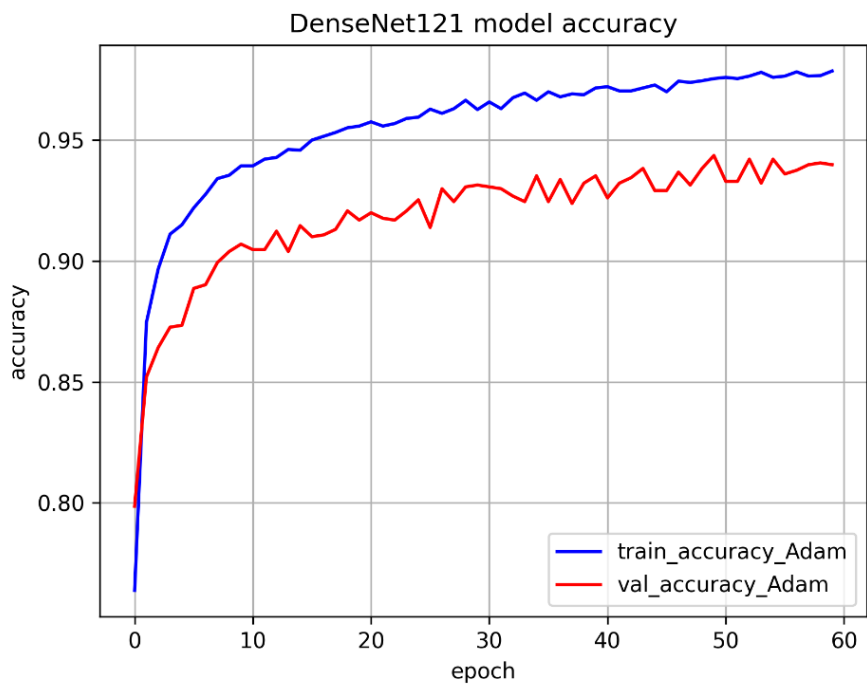
Slika 35. Gubitak ResNet50 modela koristeći dropout(0.5) kod RMSprop optimizacijskog algoritma

Budući da je i u DenseNet121 arhitekturi korišten predtrenirani model na ImageNet skupu podataka treniran je samo zadnji sloj. Model je konfiguriran s bitnim hiperparametrima kao što je optimizacijski algoritam Adam, SGD i RMSprop te sa stopom učenja od 0.001 i epohama 100. U slučaju DenseNet121 arhitekture, treniranje je prekinuto na sljedećem broju epoha: Adam – 49, SGD –78 i RMSprop – 35. U tablici 10. prikazane su izračunate vrijednosti evaluacijskih metrika za svaki optimizacijski algoritam.

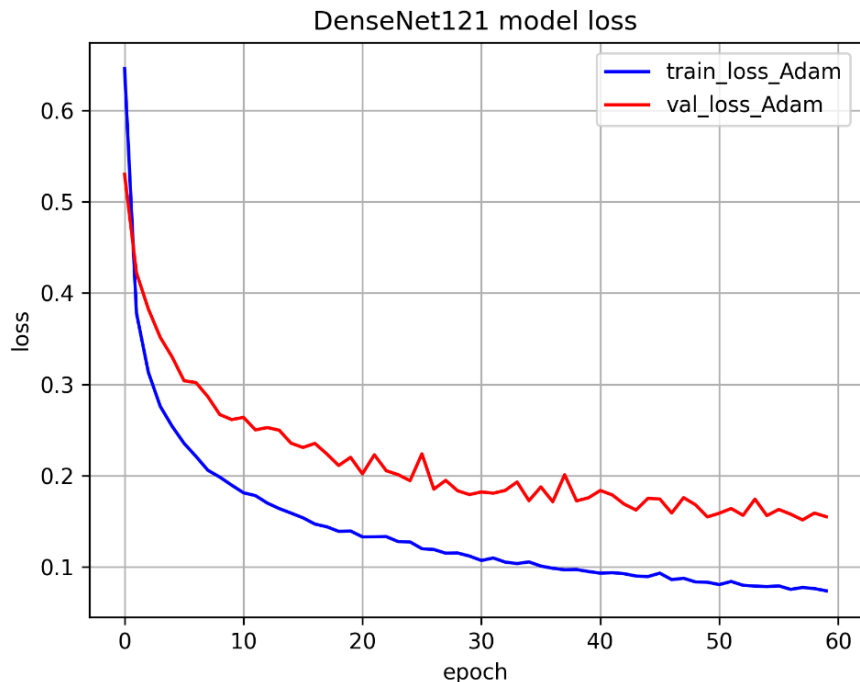
Tablica 10. Performanse DenseNet121 arhitekture za različite optimizacijske algoritme

Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1	Overfitting
Adam	0,9436	0,9404	0,9397	0,9399	Da
SGD	0,8856	0,8834	0,8804	0,8808	Da
RMSprop	0,9329	0,9310	0,9282	0,9293	Da

Prema tablici 10. moguće je vidjeti kako se Adam optimizacijski algoritam pokazao najbolji s rezultatom točnosti od 94.4% u 49 epohi. Na slikama 36. i 37. prikazana je točnost i gubitak DenseNet121 modela na trening i validacijskom skupu podataka korištenjem Adam optimizacijskog algoritma, a ostale slike grafova za SGD i RMSprop optimizacijske algoritme prikazani su pod poglavljem Dodaci.



Slika 36. Točnost Densenet121 modela koristeći Adam optimizacijski algoritam



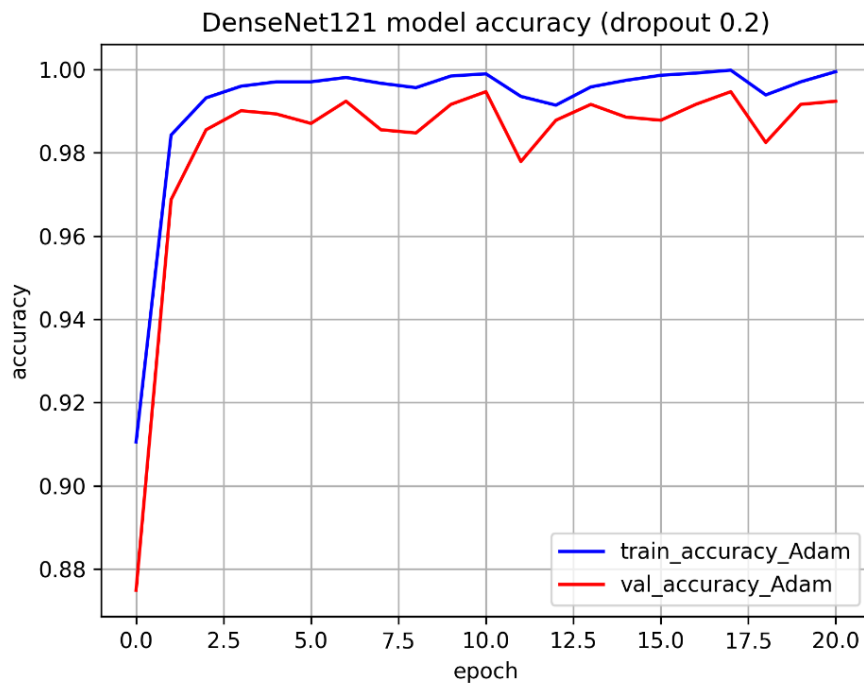
Slika 37. Gubitak DenseNet121 modela koristeći Adam optimizacijski algoritam

Kako bi se reducirao overfitting dodaje se dropout(0.2) prije izlaznog sloja. U tablici 11. su prikazani novi najbolji rezultati u 10 epohi s izmijenjenom stopom učenja 0.0001.

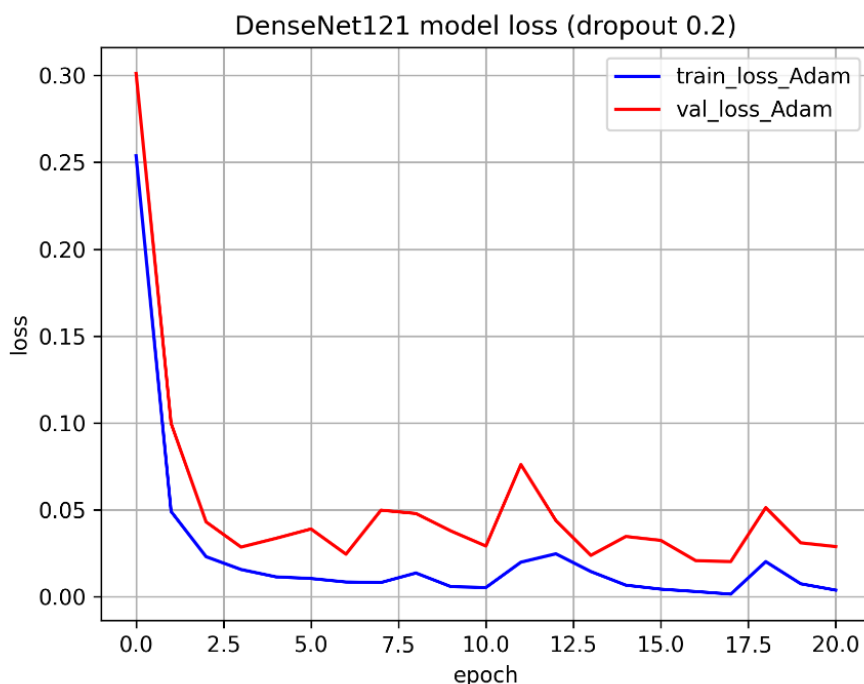
Tablica 11 Rezultati nakon dodavanja dropouth(0.2)

Optimizacijski algoritam	Točnost	Preciznost	Odaziv	F1
Adam	0.9946605645	0.9942503663	0.9941993464	0.9942152808

Na slikama 38. i 39. prikazana je točnost i gubitak DenseNet121 modela na trening i validacijskom skupu dodavanjem dropouth(0.2) za reduciranje overfittinga Adam optimizacijskog algoritma.



Slika 38. Točnost DenseNet121 modela koristeći dropout(0.2) kod Adam optimizacijskog algoritam



Slika 39. Gubitak DenseNet121 modela koristeći dropout(0.2) kod Adam optimizacijskog algoritma

Uspoređujući dobivene rezultate, DenseNet121 se pokazao kao najbolja arhitektura s najvećom točnosti, preciznosti, odazivom i F1-score. U tablici 12. prikazani su rezultati za sve arhitekture s najboljim optimizacijskim algoritmom.

Tablica 12. Performanse svih arhitekture

Arhitektura	Optimizacijski algoritmi	Točnost	Preciznost	Odaziv	F1
Prilagođena	Adam	0.985507246	0.985127562	0.984313725	0.984702264
VGG16	Adam	0.971777269	0.970214066	0.969362745	0.969615181
AlexNet	SGD	0.975591151	0.973828869	0.97362745	0.973699025
ResNet50	RMSprop	0.992372234	0.991762403	0.991797385	0.991727300
DenseNet121	Adam	0.994660564	0.994250366	0.994199346	0.994215280

Iz rezultata je vidljivo kako kompleksnije i naprednije arhitekture postižu bolje rezultate. Isto tako, naprednije arhitekture su ujedno i računalno zahtjevnije za izvođenje te se javlja i problem prilikom treniranja istih zbog veličine same mreže i velikog broja parametara koji se ugađaju. Predtrenirane modele je lakše trenirati jer je ponekad, ovisno o problemu, dovoljno trenirati samo zadnji sloj ili nekoliko zadnjih slojeva kako bi se postigli zadovoljavajući rezultati. Nije

potrebno trenirati kompletan model što uvelike smanjuje i potencijalni problem overfittinga. Prema dobivenim rezultatima, prilagođena arhitektura trenirana koristeći Adam optimizacijski algoritam postigla je bolje rezultate u odnosu na postojeće arhitekture VGG16 i AlexNet trenirane s Adam i SGD algoritmima. Međutim, DenseNet121 kao najnaprednija od navedenih arhitektura u kombinaciji s Adam optimizacijskim algoritmom, postigla je najbolje rezultate na validacijskom skupu podataka s vrijednostima točnosti, preciznosti, odaziva i F1 od 0.9947, 0.9943, 0.9942 i 0.9942.

5. ZAKLJUČAK

Detekcija i dijagnoza tumora mozga ključna je u liječenju, a pristup automatskog otkrivanja tumora na mozgu olakšava otkrivanje, ali i značajno povećava šanse pacijenata za preživljavanje. Konvolucijske neuronske mreže za kategorizaciju tumora mozga pomogle su utrti put bržem i boljem otkrivanju tumora. Takav koncept može detektirati značajke na slikama koristeći Tensorflow izgradnjom umjetne konvolucijske neuronske mreže. Ovaj rad je komparativna analiza različitih optimizacijskih algoritama korištenih u različitim arhitekturama CNN-a za izvođenje i procjenu izvedbe detekcije i klasifikacije tumora mozga u četiri klase. Analogija je napravljena na javno dostupnom setu slika mozga magnetskom rezonancom. Od svih korištenih arhitektura, najbolje se pokazala DenseNet121 koristeći prethodno trenirani model s Adam optimizacijskim algoritmom te dodavanjem dropouta(0.2) prije izlaznog sloja radi reduciranja overfittinga. I kvantitativni i grafički rezultati pokazuju da svi optimizacijski algoritmi rade dosljedno, ali Adam radi brže s točnošću od 99.5% u desetoj epohi. Najlošije se pokazala VGG16 arhitektura s SGD optimizacijskim algoritmom s točnošću od 93.3% u sedamdeset trećoj epohi koristeći prethodno trenirani model i dodavanjem dropouta(0.5) radi reduciranja overfittinga. CNN je osmišljen da minimizira ili ponekad opozove korake prethodne obrade i obično se koristi za obradu neobrađenih slika. Naprednije arhitekture obično postižu bolje rezultate međutim sama kompleksnost treniranja također je veća te su potrebne i veće računalne performanse.

6. LITERATURA

- [1] Kavitha, A.R.; Chitra, L.; Kanaga, R. Brain tumor segmentation using genetic algorithm with SVM classifier.
- [2] Abiwinanda, N.; Hanif, M.; Hesaputra, S.T.; Handayani, A.; Mengko, T.R. Brain tumor classification using convolutional neural network. In Proceedings of the World Congress on Medical Physics and Biomedical Engineering 2018, Prague, Czech Republic.
- [3] S. Bacheera, M. S. S. Ram, Classification of brain tumors using deep features extracted using CNN, J. Phys.
- [4] Rehman A, Naz S, Razzak MI, Akram F, Imran M (2020) A deep learning-based framework for automatic brain tumors classification using transfer learning. <https://doi.org/10.1007/s00034-019-01246-3>
- [5] Chelghoum R, Ikhlef A, Hameurlaine A, Jacquir S (2020) Transfer learning using convolutional neural network architectures for brain tumor classification from MRI images. In: Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial intelligence applications and innovations AIAI 2020 IFIP advances in information and communication technology. Springer, Newyork
- [6] S interneta: "General Information About Adult Brain Tumors", Pristupljeno: 14/8/2022.
- [7] S interneta: "Adult Brain Tumors Treatment", Pristupljeno: 14/8/2022.
- [8] S interneta: <https://www.cancerresearchuk.org/about-cancer/brain-tumours/about>, Pristupljeno: 14/8/2022.
- [9] S interneta: "Mood Swings and Cognitive Changes | American Brain Tumor Association", Pristupljeno: 14/8/2022.
- [10] S interneta: www.brainsciencefoundation.org., Pristupljeno: 15/8/2022.
- [11] S interneta: <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084>, Pristupljeno: 16/8/2022.
- [12] S interneta: <https://miamineurosciencecenter.com/en/conditions/brain-tumors/types/#glioma>, Pristupljeno: 18/8/2022.
- [13] S interneta: <https://mayfieldclinic.com/pe-braintumor.htm>, Pristupljeno: 18/8/2022.
- [14] Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook.
- [15] David Baillot, Struktura neurona, <https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html>, Pristupljeno: 23/8/2022.

- [16] Akshay L Chandra, McCulloch-Pitts Neuron — Mankind’s First Mathematical Model Of A Biological Neuron, Published in Towards Data Science.
- [17] S interneta:<https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>, Pristupljeno: 23/8/2022.
- [18] CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning, Aravindpai Pai — Published On February 17, 2020.
- [19] S interneta:<https://www.javatpoint.com/artificial-neural-network>, Pristupljeno: 25/8/2022.
- [20] S interneta: <https://www.tibco.com/reference-center/what-is-a-neural-network>, Pristupljeno: 25/8/2022.
- [21] S interneta:<https://www.geeksforgeeks.org/activation-functions-neural-networks/>, Pristupljeno: 25/8/2022.
- [22] Everything you need to know about “Activation Functions” in Deep learning models, Published in Towards Dana Science.
- [23] Fundamentals of Deep Learning – Activation Functions and When to Use Them?, Dishashree26 Gupta — Published On January 30, 2020 and Last Modified On July 19th, 2020
- [24] S interneta: <https://medium.com/swlh/activation-functions-in-artificial-neural-networks-8aa6a5ddf832>, Pristupljeno: 30/8/2022.
- [25] S interneta: <https://www.upgrad.com/blog/types-of-activation-function-in-neural-networks/>, Pristupljeno: 30/8/2022.
- [26] Serafeim Loukas, What is Machine Learning: Supervised, Unsupervised, Semi-Supervised and Reinforcement learning methods, Published in Towards Data Science.
- [27] Haykin, S. (2009). Neural Networks and Learning Machines. Third Edition. PEARSON, Prentice Hall
- [28] Vitaly Bushaev, Adam — latest trends in deep learning optimization, Published in Towards Dana Science.
- [29] S interneta: <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html#>
- [30] Bottou, Léon; Bousquet, Olivier (2012). "The Tradeoffs of Large Scale Learning". In Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (eds.). Optimization for Machine Learning. Cambridge: MIT Press..
- [31] A Comprehensive Guide on Deep Learning Optimizers, Published as a part of the Data Science Blogathon.

- [32] S interneta: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>, Pristupljeno: 15/9/2022.
- [33] S interneta: <https://ch.mathworks.com/discovery/convolutional-neural-network-matlab.html>, Pristupljeno: 15/9/2022.
- [34] Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition by Ian Goodfellow, Yoshua Bengio, Aaron Courville.
- [35] Jason Brownlee on April 17, 2019 in Deep Learning for Computer Vision, How Do Convolutional Layers Work in Deep Learning Neural Networks?
- [36] S interneta: <https://androidkt.com/explain-pooling-layers-max-pooling-average-pooling-global-average-pooling-and-global-max-pooling/>, Pristupljeno: 4/9/2022.
- [37] Rohit Thakur, Step by step VGG16 implementation in Keras for beginners, Published in Towards Dana Science.
- [38] Vihar Kurama, A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet.
- [39] S interneta: <https://blog.devgenius.io/resnet50-6b42934db431>, Pristupljeno: 11/9/2022.
- [40] Sabyasachi Sahoo, Residual blocks — Building blocks of ResNet, Published in Towards Dana Science
- [41] Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images, Qingge Ji 1,2, Jie Huang 1,2, Wenjie He 1,2 and Yankui Sun 2,3
- [42] S interneta: <https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow>, Pristupljeno: 11/9/2022.
- [43] Shweta Goyal , Evaluation Metrics for Classification Models, Published in Analytics Vidhya
- [44] Koo Ping Shung , Accuracy, Precision, Recall or F1?, Published in Towards Dana Science.
- [45] Sasikala, M., and N. Kumaravel. Wavelet based automatic segmentation of brain tumors using optimal texture features. 4th Kuala Lumpur International Conference on Biomedical Engineering 2008. Springer, Berlin, Heidelberg, 2008.
- [46] Mathur, Neha, et al. Detection of Brain Tumor in MRI Image through Fuzzy-Based Approach. High-Resolution Neuroimaging-Basic Physical Principles and Clinical Applications.

7. POPIS SLIKA

Slika 1. Lokacija tumora mozga [13].....	5
Slika 2. Primjer tumora mozga za svaku klasu.....	6
Slika 3. Biološka neuronske mreža [19].....	8
Slika 4. Umjetna neuronska mreža [19].....	8
Slika 5. Arhitektura umjetne neuronske mreže [20].....	9
Slika 6. Sigmoidna funkcija [21].....	11
Slika 7. Hiperbolička tangentna funkcija [21].....	12
Slika 8. ReLU funkcija [21].....	13
Slika 9. Učenje s učiteljem [27].....	14
Slika 10. Učenje bez učitelja [27].....	15
Slika 11. Ojačano učenje [27].....	15
Slika 12. Primjer osnovne arhitekture konvolucijske neuronske mreže [33].....	19
Slika 13. Filtar primijenjen na dvodimenzionalni ulaz za izradu mape značajki [35].....	20
Slika 14. Primjer max poolinga [36].....	21
Slika 15. Primjer average-poolinga [36].....	22
Slika 16. Arhitektura VGG16 modela [37].....	24
Slika 17. Arhitektura AlexNet modela [38].....	26
Slika 18. Residual blok [40].....	28
Slika 19. Arhitektura ResNet50 [41].....	29
Slika 20. Arhitektura DenseNet121 [41].....	31
Slika 21. Prilagođena arhitektura.....	32
Slika 22. Točnost prilagođenog modela koristeći Adam optimizacijski algoritam.....	36
Slika 23. Gubitak prilagođenog modela koristeći Adam optimizacijski algoritam.....	36
Slika 24. Točnost prilagođenog modela koristeći dropout(0.5) kod Adam optimizacijskog algoritam.....	37
Slika 25. Gubitak prilagođenog modela koristeći dropout(0.5) kod Adam optimizacijskog algoritma.....	38
Slika 26. Točnost VGG16 modela koristeći Adam optimizacijski algoritam.....	39
Slika 27. Gubitak VGG16 modela koristeći Adam optimizacijski algoritam.....	39
Slika 28. Točnost VGG16 modela koristeći dropout(0.5) kod Adam optimizacijskog algoritam.....	40
Slika 29. Gubitak VGG16 modela koristeći dropout(0.5) kod Adam optimizacijskog algoritma.....	41
Slika 30. Točnost AlexNet modela koristeći SGD optimizacijski algoritam.....	42
Slika 31. Gubitak AlexNet modela koristeći SGD optimizacijski algoritam.....	43
Slika 32. Točnost ResNet50 modela koristeći RMSprop optimizacijski algoritam.....	44
Slika 33. Gubitak ResNet50 modela koristeći RMSprop optimizacijski algoritam.....	45
Slika 34. Točnost ResNet50 modela koristeći dropout(0.5) kod RMSprop optimizacijskog algoritma.....	46
Slika 35. Gubitak ResNet50 modela koristeći dropout(0.5) kod RMSprop optimizacijskog algoritma.....	46
Slika 36. Točnost Densenet121 modela koristeći Adam optimizacijski algoritam.....	48
Slika 37. Gubitak DenseNet121 modela koristeći Adam optimizacijski algoritam.....	48
Slika 38. Točnost DenseNet121 modela koristeći dropout(0.2) kod Adam optimizacijskog algoritam.....	49
Slika 39. Gubitak DenseNet121 modela koristeći dropout(0.2) kod Adam optimizacijskog algoritma.....	50

8. POPIS TABLICA

Tablica 1. Distribucija baze podataka u četiri klase.....	6
Tablica 2. Odnos biološke neuronske mreže i umjetne neuronske mreže	9
Tablica 3. Performanse prilagođene arhitekture za različite optimizacijske algoritme	35
Tablica 4. Rezultati nakon dodavanja dropouta(0.5)	37
Tablica 5. Performanse VGG16 arhitekture za različite optimizacijske algoritme	38
Tablica 6. Rezultati nakon dodavanja dropouta(0.5)	40
Tablica 7. Performanse AlexNet arhitekture za različite optimizacijske algoritme.....	41
Tablica 8. Performanse ResNet50 arhitekture za različite optimizacijske algoritme	43
Tablica 9. Rezultati nakon dodavanja dropouta(0.5)	45
Tablica 10. Performanse DenseNet121 arhitekture za različite optimizacijske algoritme	47
Tablica 11 Rezultati nakon dodavanja dropouta(0.2)	49
Tablica 12. Performanse svih arhitekture	50

9. SAŽETAK

Tumor mozga skup je abnormalnih stanica u mozgu koje oštećuju zdrave moždane stanice. Dijagnoza i utvrđivanje točne vrste i stupnja tumora u ranim stadijima igraju važnu ulogu u odabiru odgovarajućeg plana liječenja. Nedavni napredak u području dubokog učenja pomogao je zdravstvenoj industriji u dijagnosticiranju mnogih bolesti. Konvolucijska neuronska mreža (CNN) najrašireniji je i najčešće korišten algoritam za vizualno učenje i prepoznavanje slika. Predloženi rad uključuje pristup duboke neuronske mreže i modele VGG16, AlexNet, ResNet50, DenseNet121 i prilagođenu arhitekturu temeljenu na CNN-u za klasificiranje magnetske rezonancije (MRI) u četiri kategorije. Duboke mreže zajedno s prijenosnim učenjem treniraju se i testiraju korištenjem tri različita optimizacijska algoritma (Adam, SGD i RMSprop) koristeći javno dostupne podatke koji se sastoje od 7022 MRI slike. Predloženi sustavi postigli su zadovoljavajuće rezultate u pogledu višerazredne klasifikacije.

Ključne riječi: Umjetna inteligencija, tumor mozga, klasifikacija, konvolucijska neuronska mreža, duboko učenje.

ABSTRACT

A brain tumor is a collection of abnormal cells in the brain that damage healthy brain cells. Diagnosis and identification of the exact type and stage of the tumor in the early stages play an important role in choosing the appropriate treatment plan. Recent advances in the field of deep learning have helped the healthcare industry in the diagnosis of many diseases. A convolutional neural network (CNN) is the most widespread and most used algorithm for visual learning and

IZJAVA

Sukladno članku 9. Pravilnika o završnom radu, završnom ispitu i završetku sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci izjavljujem da sam samostalno izradila diplomski rad koristeći znanja stečena tijekom studiranja na Tehničkom fakultetu u Rijeci te uz korištenje navedene literature.

Natalija Ivoš

Ivoš

IZJAVA

Sukladno članku 9. Pravilnika o završnom radu, završnom ispitu i završetku sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci izjavljujem da sam samostalno izradila diplomski rad koristeći znanja stečena tijekom studiranja na Tehničkom fakultetu u Rijeci te uz korištenje navedene literature.

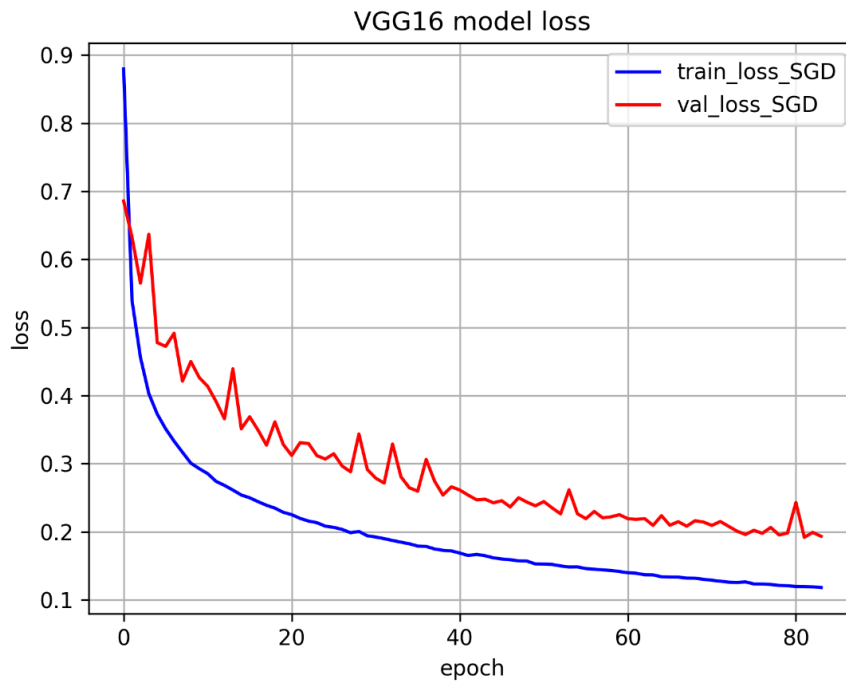
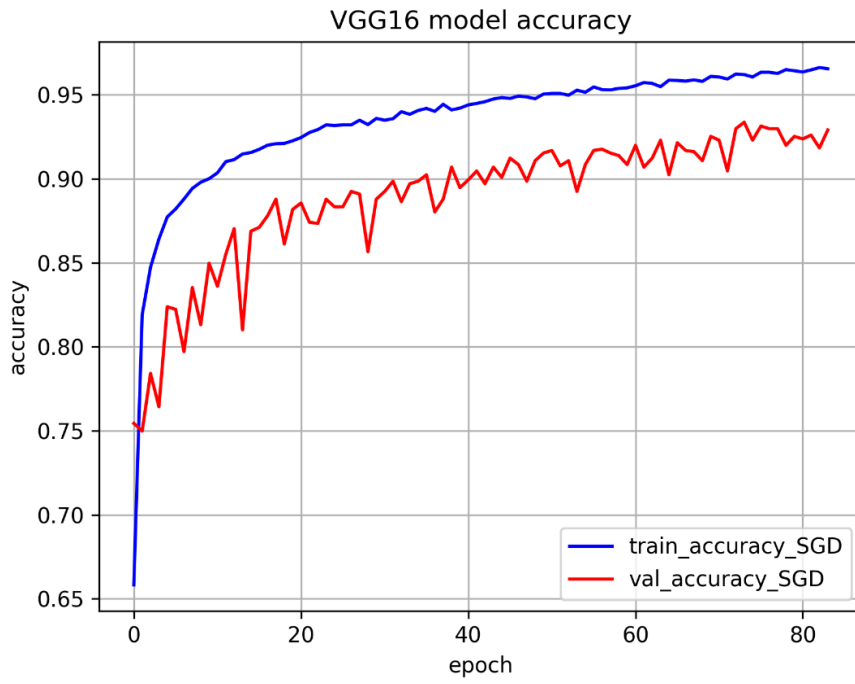
Natalija Ivoš

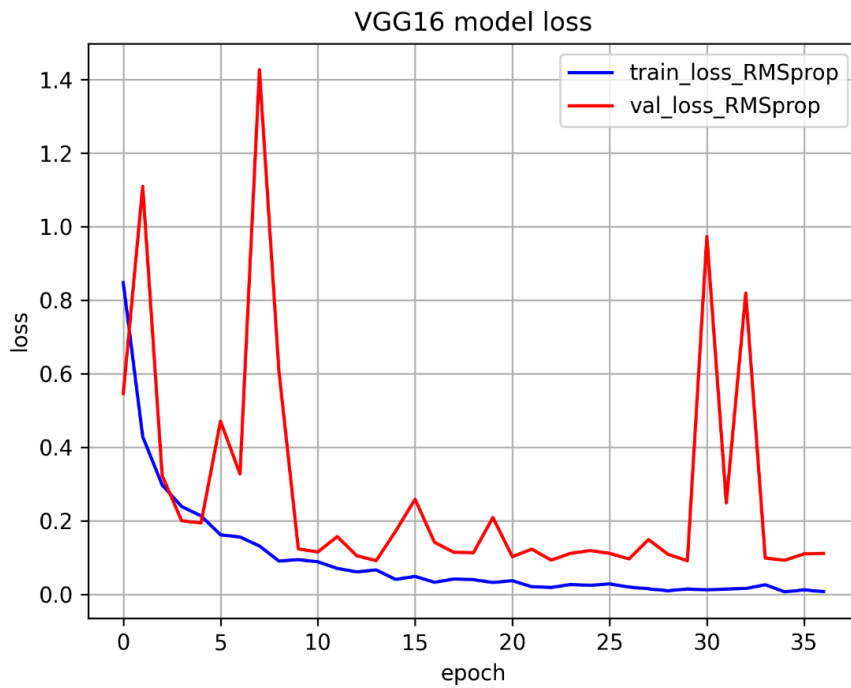
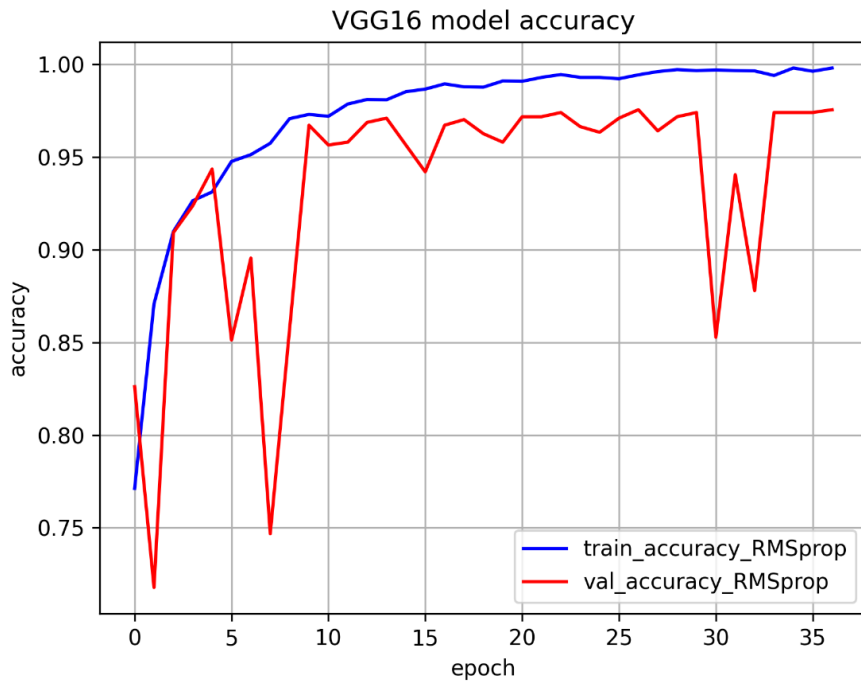
Ivoš

image recognition. The proposed work includes a deep neural network approach and models VGG16, AlexNet, ResNet50, DenseNet121 and a custom CNN-based architecture to classify magnetic resonance imaging (MRI) into four categories. Deep transfer learning networks are trained and tested using three different optimization algorithms (Adam, SGD and RMSprop) on publicly available dataset containing 7022 MRI images. The proposed systems achieved satisfactory results in terms of multi-class classification.

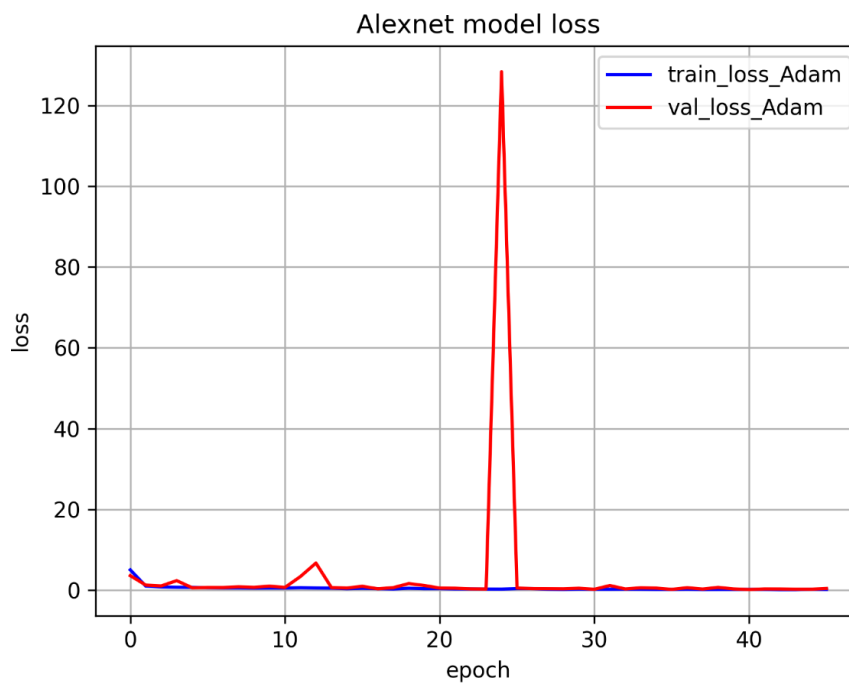
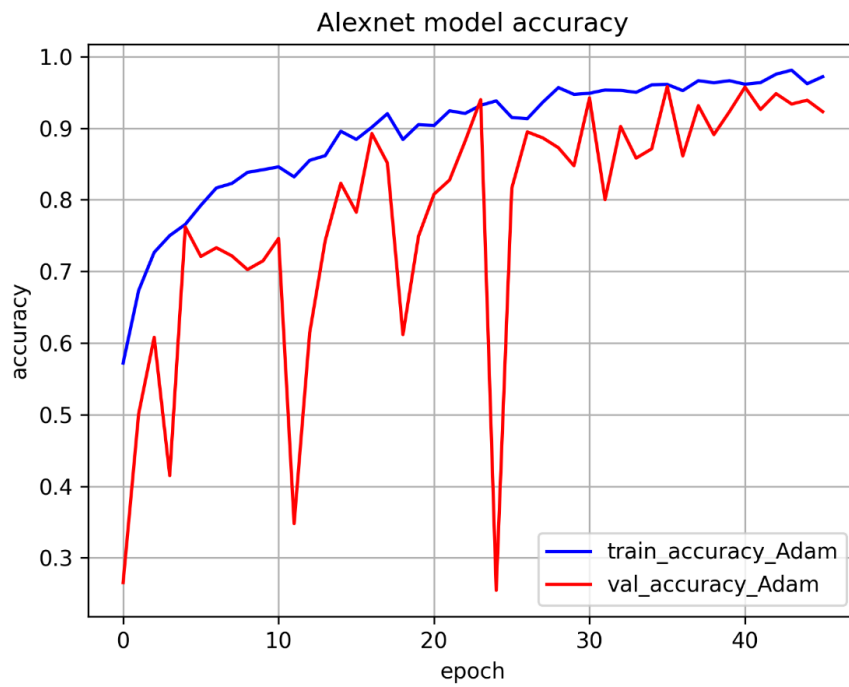
Keywords: Artificial intelligence, brain tumor, classification, convolutional neural network, deep learning.

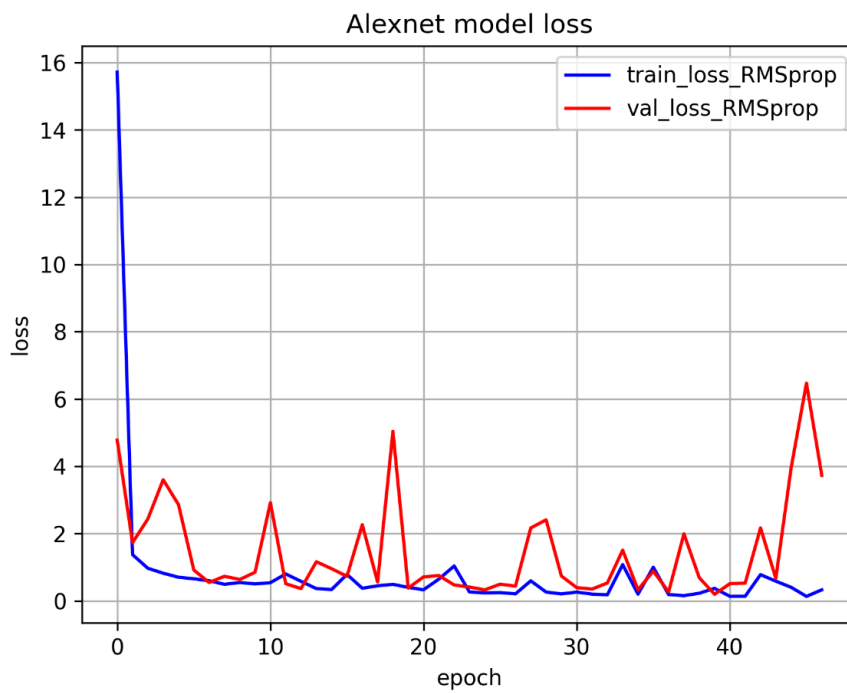
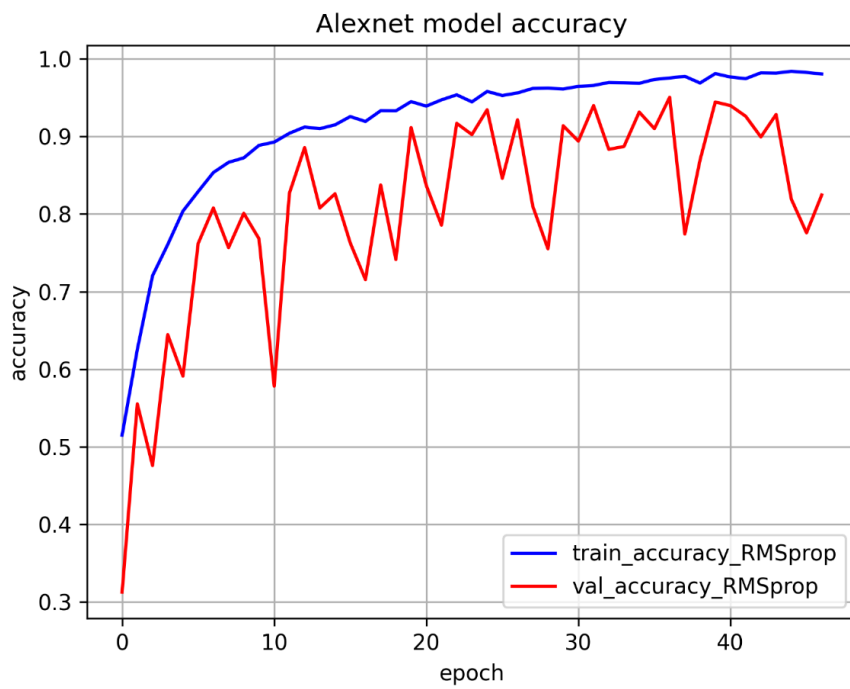
DODATAK A - Grafovi točnosti i funkcije gubitka za VGG16 model



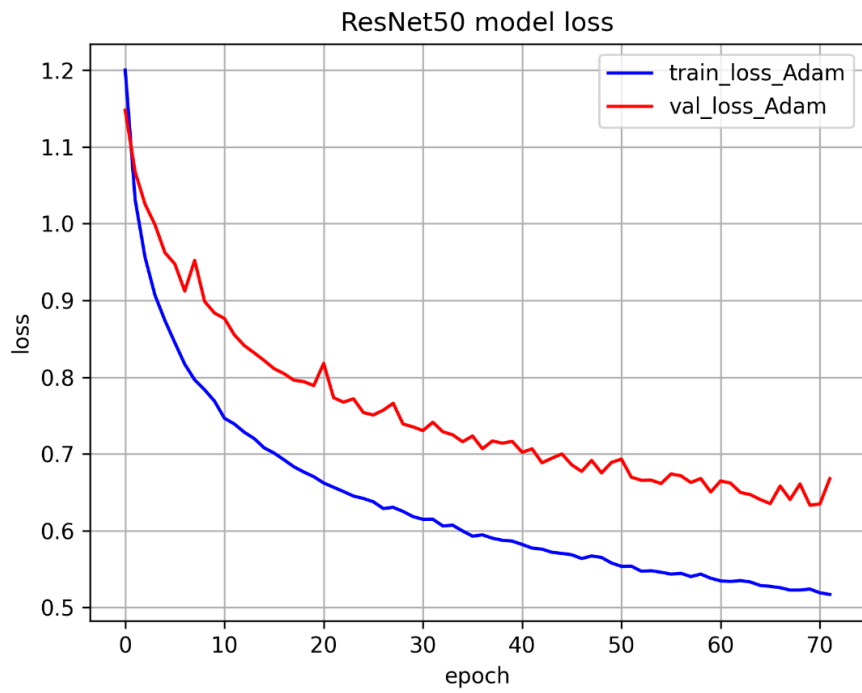
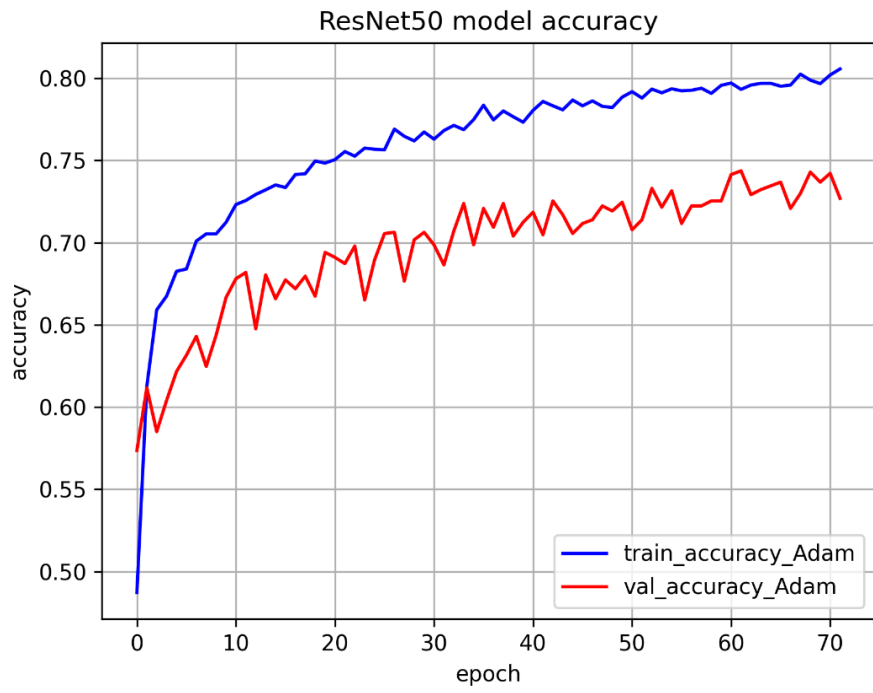


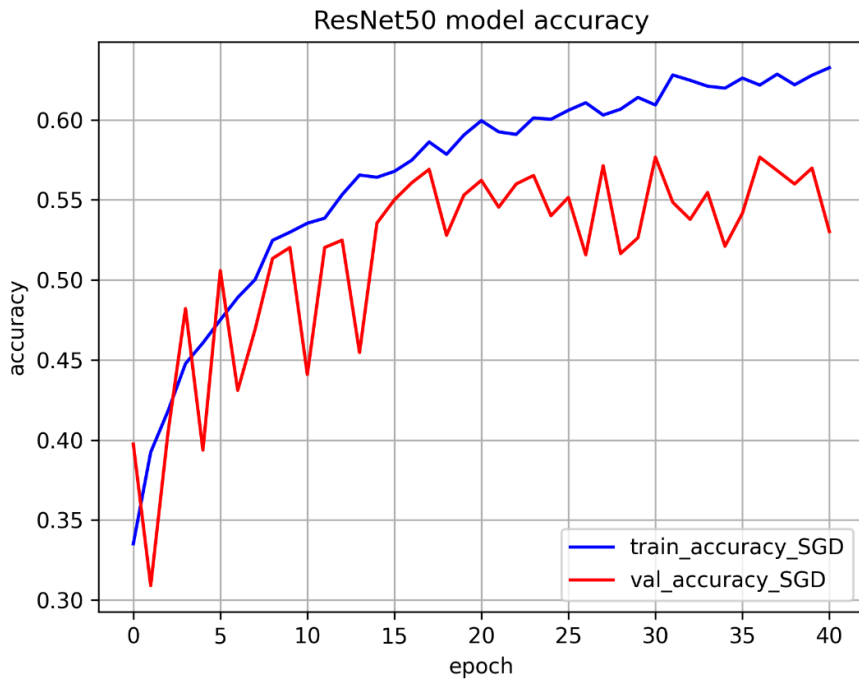
DODATAK B - Grafovi točnosti i funkcije gubitka za AlexNet model



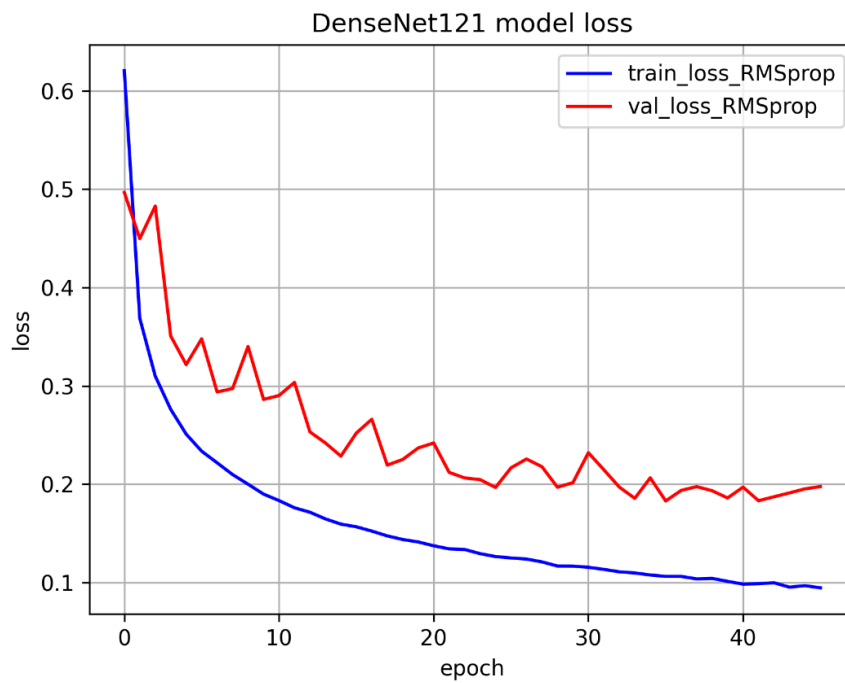
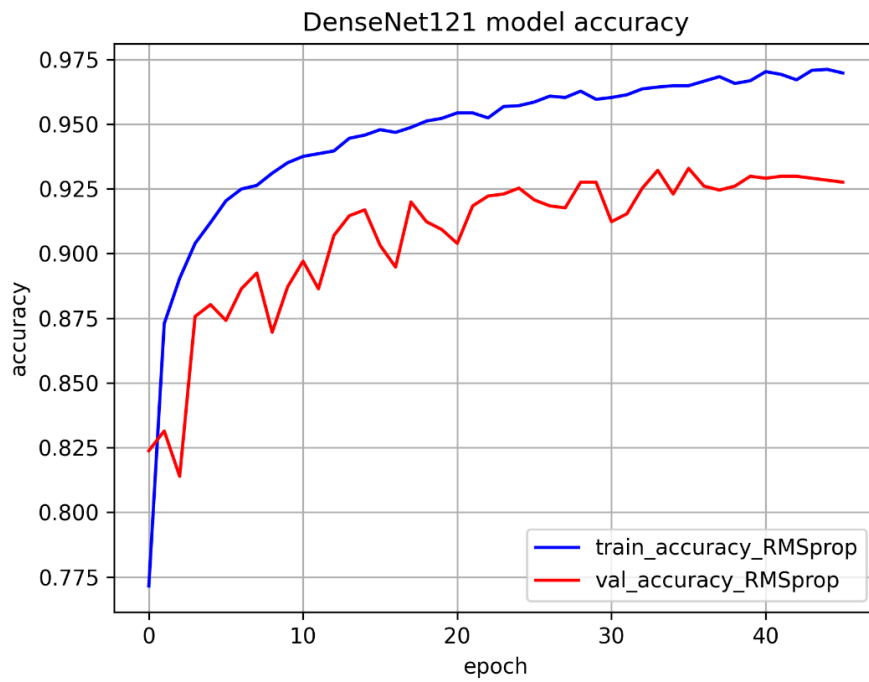


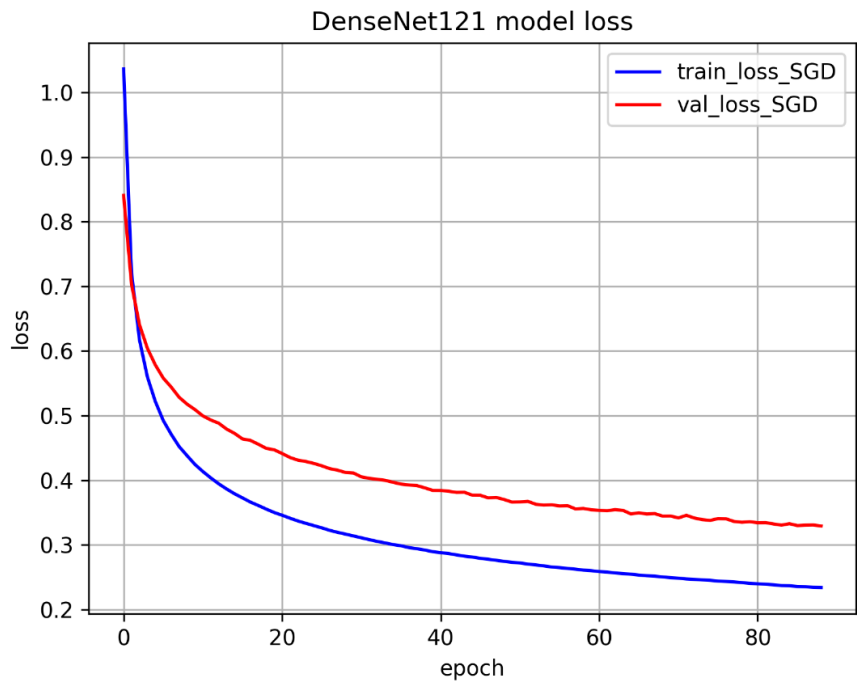
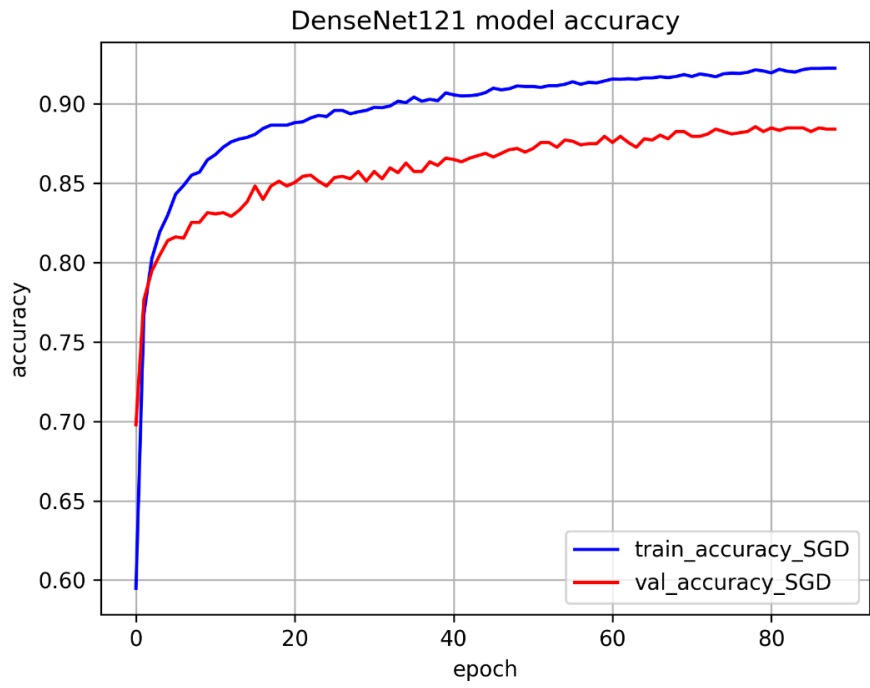
DODATAK C - Grafovi točnosti i funkcije gubitka za ResNet50 model





DODATAK D - Grafovi točnosti i funkcije gubitka za DenseNet121 model





DODATAK E - Grafovi točnosti i funkcije gubitka za prilagođeni model

