

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2813

**DETEKCIJA UPALE PLUĆA NA RENDGENSKIM
SNIMKAMA UPORABOM METODA DUBOKOG UČENJA**

Borna Popović

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2813

**DETEKCIJA UPALE PLUĆA NA RENDGENSKIM
SNIMKAMA UPORABOM METODA DUBOKOG UČENJA**

Borna Popović

Zagreb, lipanj 2022.

DIPLOMSKI ZADATAK br. 2813

Pristupnik: **Borna Popović (0036494870)**

Studij: Računarstvo

Profil: Računarska znanost

Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Detekcija upale pluća na rendgenskim snimkama uporabom metoda dubokog učenja**

Opis zadatka:

Upala pluća respiratorna je bolest koju uzrokuju bakterije ili virusi. Najčešće zahvaća stanovnike manje razvijenih područja s višom razinom zagađenja, lošijim higijenskim uvjetima i nižim životnim standardima. Rana dijagnoza upale pluća jako je bitna jer znatno smanjuje stopu smrtnosti. Analiza rendgenskih snimaka najčešći je način dijagnosticiranja upale pluća. Analiza tih snimaka dosta je zahtjevna i često dosta ovisi o subjektivnoj procjeni stručnjaka. U ovom diplomskom radu razmatrat će se različiti modeli dubokog učenja koji omogućuju detekciju upale pluća na temelju rendgenskih snimaka. Rezultati razvijene konvolucijske neuronske mreže uspoređivat će se s raznim modelima korištenim u drugim radovima dostupnim na internetu kako bi se ustanovili koji modeli daju najbolje rezultate na dostupnom skupu podataka. Za učenje i ispitivanje modela koristit će se programski jezik python i biblioteka keras. Za razvojnu okolinu koristit će se Google Colab zbog jednostavnog pristupa GPU-u, što značajno ubrzava process učenja modela. Slobodno dostupni skup podataka koji će se koristiti za učenje i testiranje modela sastoji se od oko 5000 rendgenskih snimaka koje zauzimaju više od 2 GB memorijskog prostora.

Rok za predaju rada: 27. lipnja 2022.

Zahvaljujem svom mentoru, izv. prof. dr. sc. Alanu Joviću, na stručnom vodstvu, pomoći i strpljenju te izdvojenom vremenu za odgovaranje na sve moje upite prilikom izrade ovog diplomskog rada.

Puno hvala cijeloj mojoj obitelji i djevojci Patriciji na potpori tijekom života i cjelokupnog studiranja.

*Mentor: izv. prof. dr. sc. Alan Jović
Broj stranica: 40*

Sadržaj

1. Uvod	1
2. Duboko učenje.....	2
3. Konvolucijska neuronska mreža	3
3.1 Elementi konvolucijske neuronske mreže	4
3.1.1 Konvolucijski sloj.....	4
3.1.2 Sloj sažimanja	9
3.1.3 Sloj izravnavanja	11
3.1.4 Potpuno povezani sloj	12
3.2 Učenje konvolucijske neuronske mreže	13
4. Skup podataka.....	15
5. Tehnike evaluacije performansi modela.....	17
6. Implementacija konvolucijske neuronske mreže	20
6.1 Predobrada skupa podataka.....	20
6.2 Model konvolucijske neuronske mreže	22
6.3 Prevođenje i prilagodba modela.....	25
6.4 Rezultati naučene neuronske mreže	27
7. Rezultati i metode iz drugih radova	30
8. Usporedba rezultata s drugim radovima	34
9. Zaključak.....	35
10. Literatura.....	36
Sažetak.....	39
Summary	40

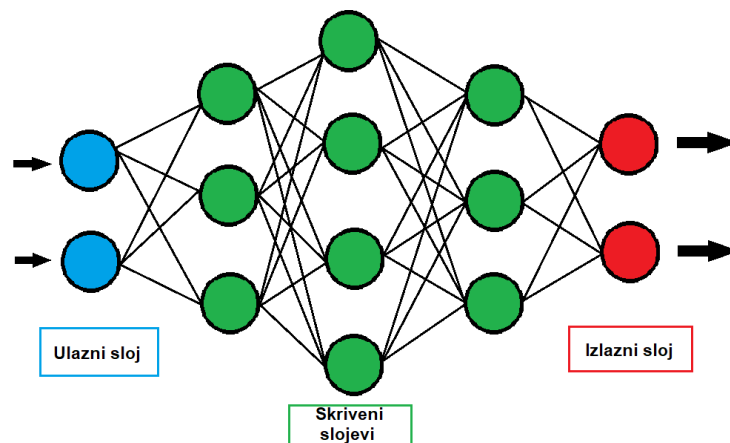
1. Uvod

Upala pluća je respiratorna bolest koju uzrokuju bakterije ili virusi. Uzročnici bolesti napadaju alveole, šupljine u plućima gdje dolazi do izmjene plinova. Posljedično dolazi do punjenja alveola tekućinom ili gnojem što dovodi do otežanog disanja (Normandin, 2021). Upala pluća najčešće zahvaća stanovnike manje razvijenih područja s višom razinom zagađenja, lošijim higijenskim uvjetima i nižim životnim standardima. Svake godine više od 150 milijuna ljudi razvije ovu bolest (Stephen, 2019). Ranom dijagnozom upale pluća moguće je znatno smanjenje stope smrtnosti. Najčešći način dijagnosticiranja upale pluća je analiza rendgenskih snimaka. Analiza tih snimaka dosta je zahtjevna i često dosta ovisi o subjektivnoj procjeni stručnjaka. Snimke oboljelih pacijenata često su vrlo nejasne što može dovesti do pogrešne dijagnoze. Pogrešna dijagnoza i neadekvatna terapija mogu pogoršati stanje pacijenta. Veliki problem predstavlja i manjak radiologa u slabo razvijenim zemljama gdje je i najveći broj zaraženih ljudi. Zbog sve većeg broja oboljelih pacijenata, a samim time i više rendgenskih snimaka prsnog koša pojavljuje se potreba za razvojem sustava za računalnu dijagnostiku (*CAD system – Computer-aided diagnosis system*) (Rajaraman, 2018). Tehnike dubokog učenja sve više se uspješno primjenjuju u području medicinskog snimanja. Na taj način se analiziraju i klasificiraju medicinske snimke u svrhu detekcije raznih bolesti kao što su rak kože, rak dojke, tuberkuloza i tumor na mozgu (Yamashita, 2018).

U ovom diplomskom radu posvetit ću se razvoju konvolucijske neuronske mreže (eng. *convolutional neural network - CNN*) koja omogućuje detekciju značajki na rendgenskim snimkama prsnog koša. Pronađene značajke koriste se za detekciju upale pluća. Ovom metodom obrade rendgenskih snimaka značajno bi se ubrzao proces analize rendgenskih snimaka te bi se povećala točnost dijagnoza u usporedbi s ručnim pristupom.

2. Duboko učenje

Duboko učenje je podskup strojnog učenja, a u većini slučajeva predstavlja neuronsku mrežu s više od tri skrivena sloja. Neuronske mreže pokušavaju simulirati ponašanje ljudskog mozga tako da uče i dobivaju informacije iz velikog skupa podataka. Duboko učenje je ključna tehnologija u industriji autonomnih vozila koja omogućuje prepoznavanje znakova, pješaka itd. Također se koristi i u osobnim uređajima kao što su mobilni telefoni, tableti i televizori gdje omogućuje glasovnu kontrolu (eng. *voice control*) (IBM Cloud Education, 2020). Modeli dubokog učenja uče klasificirati direktno iz slika, teksta i zvukova pa se njihova primjena sve više pojavljuje i u medicini, a u nekim slučajevima uspjeli su premašiti čovjekove performanse. O dubokom učenju teoretiziralo se još u osamdesetim godina 20. stoljeća, a tek je nedavno postalo vrlo korisno, jer zahtijeva velike količine označenih podataka koje prije nisu bile dostupne. Danas je pohrana podataka značajno napredovala i količina pohranjenih podataka raste eksponencijalno svake godine. Duboko učenje također zahtijeva veliku računalnu snagu kao što su grafičke kartice visokih performansi koje mogu raditi paralelno kako bi značajno smanjile vrijeme učenja mreže (MathWorks, 2022). Na slici 1 prikazan je primjer neuronske mreže s tri skrivena sloja.

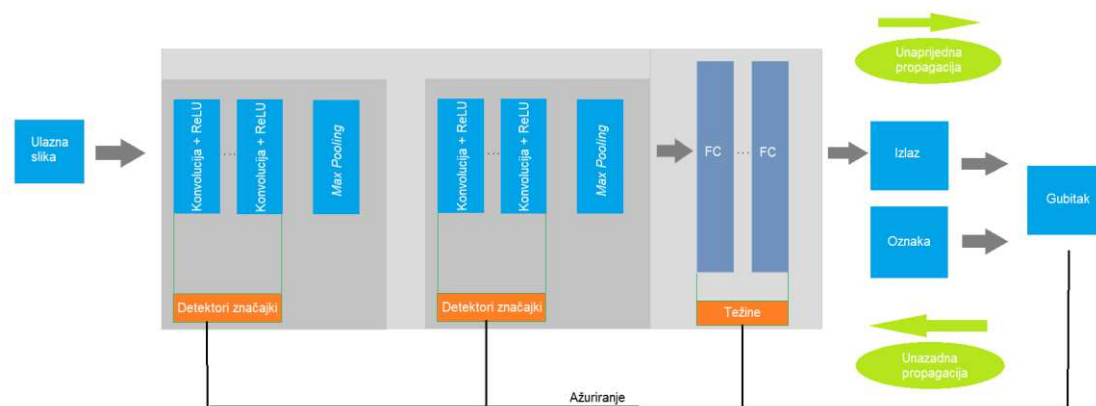


Slika 1. Neuronska mreža

3. Konvolucijska neuronska mreža

Konvolucijska neuronska mreža (eng. *convolutional neural network* - CNN) u posljednjih 10 godina sve češće pronalazi svoju uporabu u raznim područjima koja zahtijevaju raspoznavanje uzoraka obradom slike ili govora (Albawi, 2017). Jedno od tih područja je medicina gdje se sve više postižu odlični rezultati čija kvaliteta konkurira stručnjacima. CNN se sastoji od višestrukih slojeva. To su konvolucijski sloj, sloj sažimanja (eng. *pooling layer*) i potpuno povezani sloj. Oblikovana je da automatski uči i prilagođava učenje kako bi otkrila prostornu hijerarhiju značajki pomoću algoritma unazadne propagacije (eng. *backpropagation*) (Yamashita, 2018). Najveća prednost konvolucijskih neuronskih mreža u odnosu na uobičajene potpuno povezane mreže je smanjenje broja parametara umjetne neuronske mreže. Problemi koje rješavaju konvolucijske neuronske mreže ne bi trebali imati prostorno ovisne značajke. Naprimjer, ako je zadatak mreže da prepozna koja životinja se nalazi na slici, nije bitno gdje i u kojem položaju se ona nalazi na slici. Još jedna važna značajka CNN-a je otkrivanje apstraktnih značajki u dubljim slojevima mreže, gdje će se otkriti neke kompliciranije značajke specifične za pojedinu životinju, dok će se u početnim slojevima otkriti samo rubovi i jednostavniji oblici (Albawi, 2017).

Na slici 2 prikazan je primjer tipične konvolucijske neuronske mreže koja se sastoji od nekoliko blokova. Svaki od blokova sadrži nekoliko konvolucijskih slojeva, a na kraju svakog bloka je sloj sažimanja. Nakon njih slijedi jedan ili više potpuno povezanih slojeva. Mreža uči tako da ulazni podaci prolaze kroz mrežu, što nazivamo unaprijedna propagacija (eng. *forward propagation*) gdje se na kraju izračunava funkcija gubitka (eng. *loss function*). Pomoću vrijednosti dobivene u funkciji gubitka ažuriraju se težine i detektori značajki u mreži algoritmom unazadne propagacije s gradijentnim spustom kao algoritmom optimizacije.



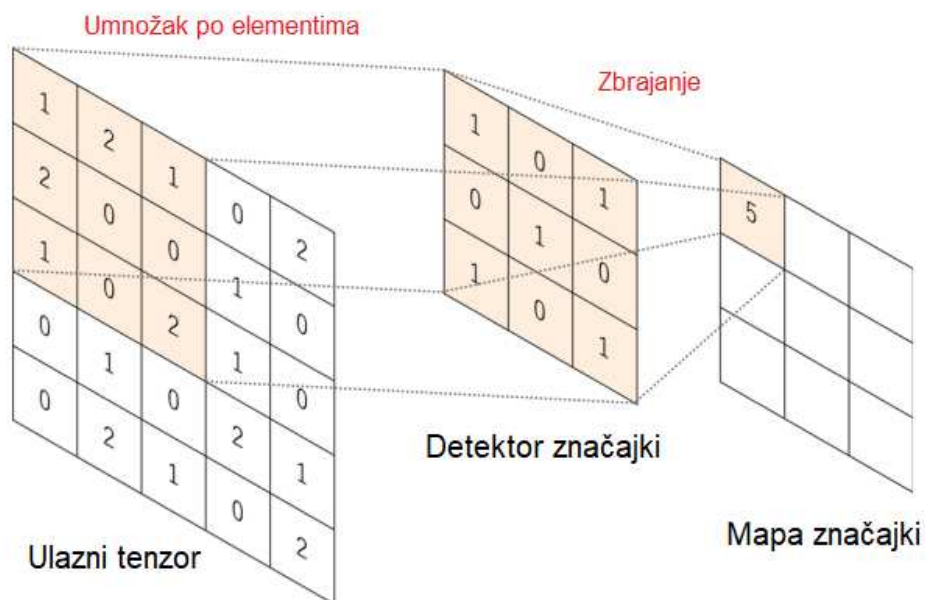
Slika 2. Pregled konvolucijske neuronske mreže (Yamashita, 2018)

3.1 Elementi konvolucijske neuronske mreže

3.1.1 Konvolucijski sloj

Konvolucijski sloj je glavna komponenta konvolucijske neuronske mreže i ima ulogu otkrivanja značajki. Koristi kombinaciju linearne operacije konvolucije i nelinearne aktivacijske funkcije (Yamashita, 2018).

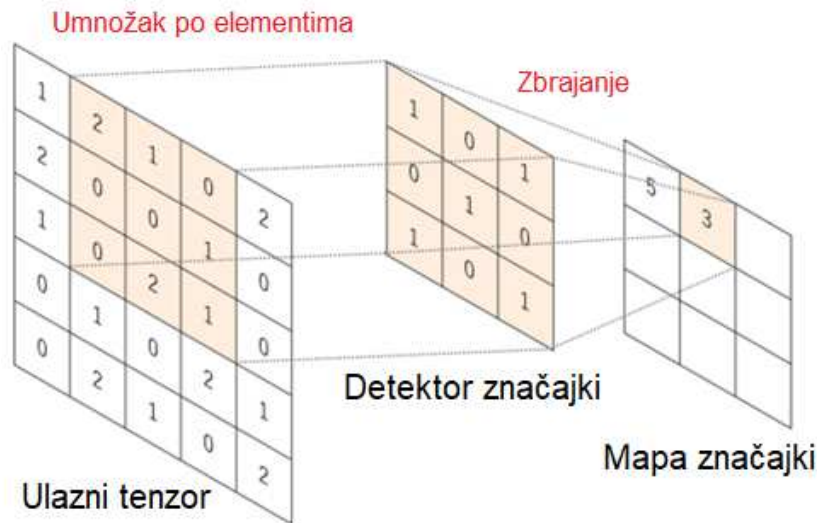
Konvolucija se koristi za pronalaženje značajki tako da se manji niz brojeva, koji nazivamo detektor značajki (eng. *feature detector*), ponekad filtar ili jezgra (eng. *kernel*), primjenjuje na ulazne podatke. Ulazni podatak je tenzor, odnosno slika s proizvoljnim brojem kanala koji su u računalu predstavljeni kao nizovi brojeva u rasponu od 0 do 255 i predstavljaju svjetlinu piksela. Ulazni tenzor se množi po elementima s detektorom značajki te se nakon toga svaki rezultat zbroji i upisuje u odgovarajuće mjesto izlaznog tenzora koji nazivamo mapa značajki (eng. *feature map*), kao što je prikazano na slici 3.



Slika 3. Konvolucija (Yamashita, 2018)

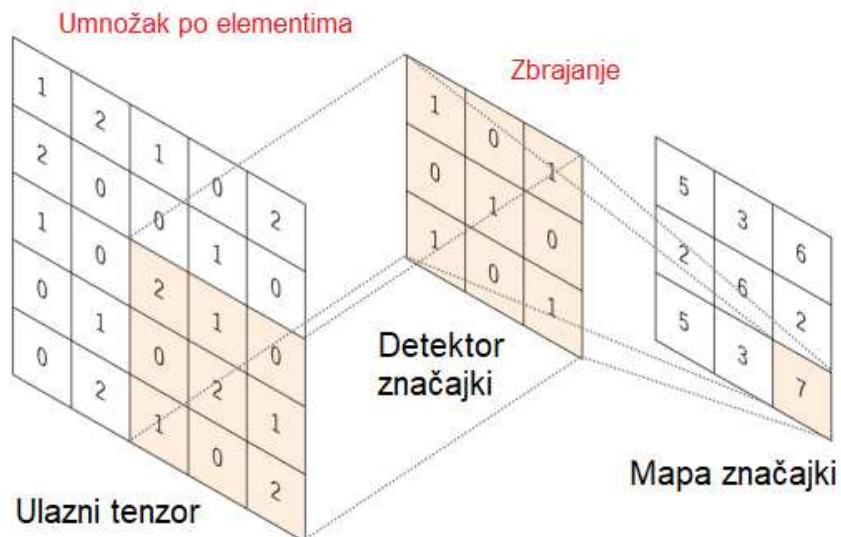
Ovaj postupak se ponavlja i primjenjuje se puno različitih detektora značajki kako bi se formirale mape značajki koje predstavljaju različite karakteristike ulaznog tenzora (Yamashita, 2018). Hiperparametri, varijable koje treba odrediti prije učenja mreže, su veličina i broj detektora značajki, korak, dopunjavanje (eng. *padding*) i aktivacijska funkcija. Parametri, varijable koje se automatski optimiraju tijekom procesa učenja, su detektori značajki.

Udaljenost između dva uzastopna položaja detektora značajki naziva se korak (eng. *stride*). Česta vrijednost koraka je 1. Veći korak se koristi kada je potrebno smanjiti uzorkovanje mape značajki (Yamashita, 2018). Na slici 4 je prikazan postupak izračunavanja pripadajuće vrijednosti mape značajki u slučaju kada je vrijednost koraka jednaka 1.



Slika 4. Konvolucija uz korak jednak 1 (Yamashita, 2018)

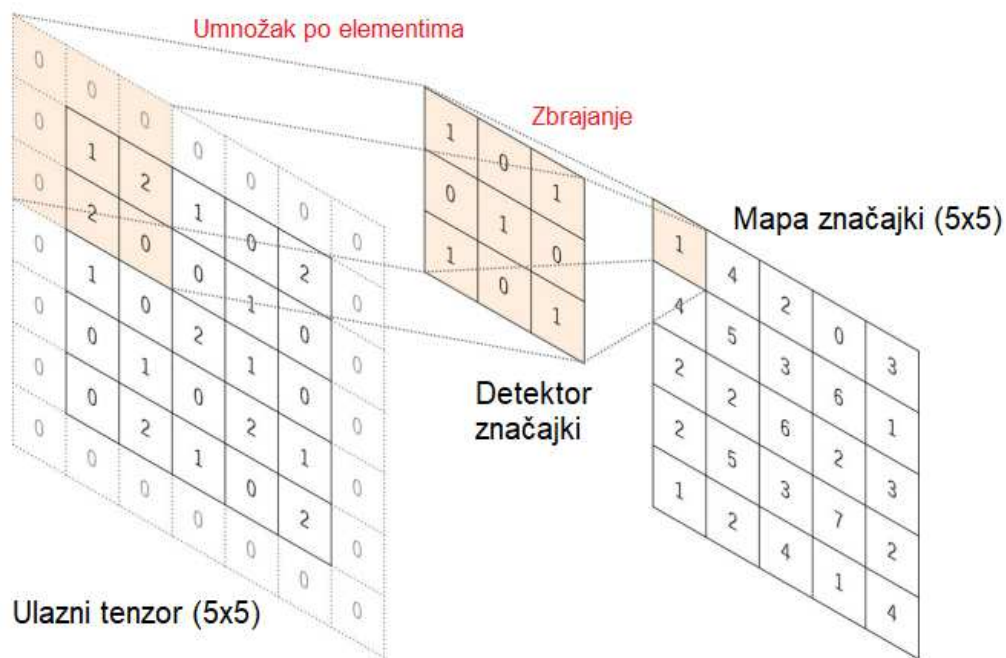
Nakon što prođemo kroz čitavi ulazni tenzor dobivamo popunjenu mapu značajki prikazanu na slici 5.



Slika 5. Popunjena mapa značajki (Yamashita, 2018)

Opisana operacija konvolucije smanjuje visinu i širinu izlazne mape značajki u usporedbi s ulaznim podacima. Također, ne postoji mogućnost da se središnji element detektora značajki preklapa s rubnim elementima ulaznog

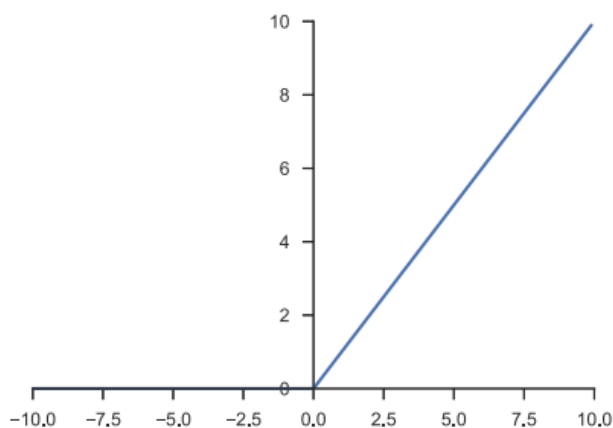
tenzora. Ovi problemi mogu se riješiti korištenjem dopunjavanja (Yamashita, 2018). Najčešće se koristi dopunjavanje nulama (eng. *zero padding*) gdje se svaki red i stupac ulaznog tenzora dopunjava nulama kako bi se omogućilo preklapanje središnjeg elementa detektora značajki s rubnim elementima ulaznog tenzora. Ovim postupkom također bi se i osigurala jednake dimenzije ulaznog tenzora i mape značajke u čitavom postupku konvolucije. Većina novijih arhitektura konvolucijske neuronske mreže koristi dopunjavanje kako bi zadržale jednake dimenzije, jer bez dopunjavanja svaka iduća konvolucija bi dodatno smanjila mapu značajki koju dobijemo na izlazu. Na slici 6 nalazi se primjer dopune nulama gdje vidimo da mapa značajki ima jednake dimenzije kao ulazni tenzor.



Slika 6. Dopunjavanje nulama (Yamashita, 2018)

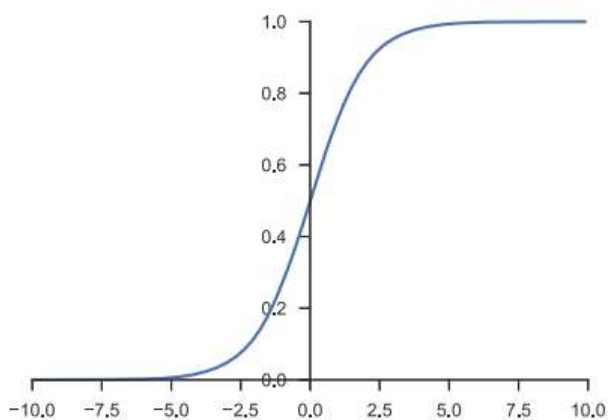
Izlaz linearne operacije konvolucije nakon toga prolazi kroz neku nelinearnu aktivacijsku funkciju. Kao aktivacijske funkcije prije su se koristile glatke nelinearne funkcije kao što su sigmoida i tangens hiperbolni, jer su one matematička reprezentacija ponašanja bioloških neurona (Yamashita, 2018).

Danas se kao aktivacijska funkcija najčešće koristi funkcija *ReLU* (tzv. zglobnica, eng. *rectified linear unit*) na čijem izlazu je ulazna vrijednost ako je riječ o pozitivnoj vrijednosti, a u suprotnom nula. Na slici 7, 8 i 9 prikazane su navedene aktivacijske funkcije i njihove formule.



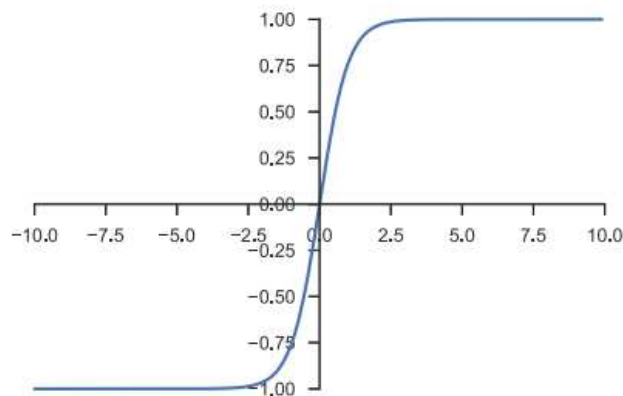
$$f(x) = \max(0, x)$$

Slika 7. Funkcija *ReLU* (Yamashita, 2018) Formula 1.



$$f(x) = \frac{1}{1 + e^{-x}}$$

Slika 8. Sigmoida (Yamashita, 2018) Formula 2.



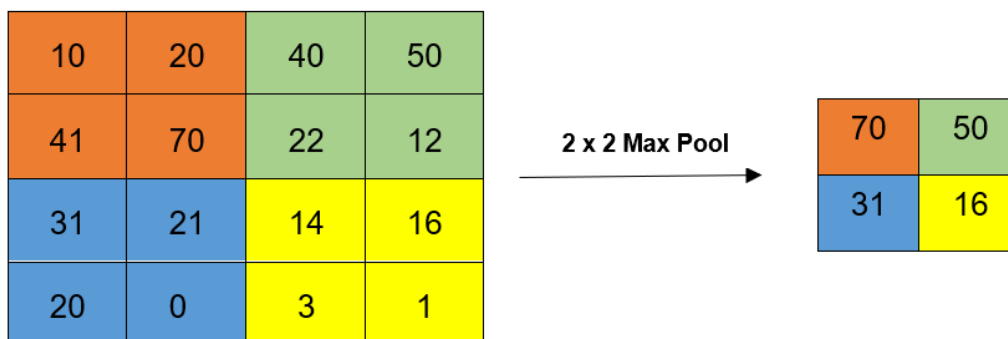
$$f(x) = \tanh(x)$$

Slika 9. Tangens hiperbolni (Yamashita, 2018) Formula 3.

3.1.2 Sloj sažimanja

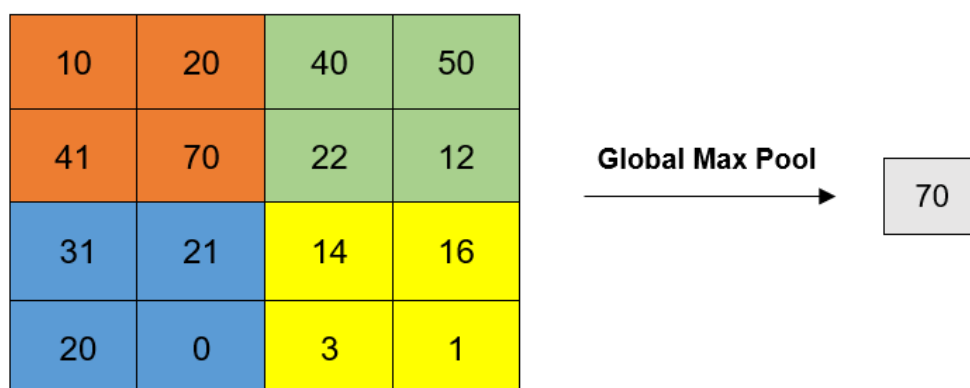
Sloj sažimanja služi kao operacija smanjenja uzorkovanja (Yamashita, 2018). Smanjuje veličinu mape značajki kako bi se uvela invarijantnost prijevoda (eng. *translation invariance*) što znači da će mreža davati istu vrijednost na izlazu neovisno o tome kako je ulazni podatak pomaknut ili izobličen. Također se smanjuje broj naknadnih parametara koji se mogu naučiti s obzirom na to da smanjujemo veličinu mape značajki. U sloju sažimanja nema parametara koji se optimiraju tijekom procesa učenja, a hiperparametri su korak, nadopuna i veličina filtra.

Najčešće korištena varijanta operacije sažimanja je maksimalno sažimanje (eng. *max pooling*) koja uzima samo maksimalnu vrijednost iz određenog dijela mape značajki. U praksi se najviše koristi maksimalno sažimanje s filtrom veličine 2x2 i korakom 2 čija primjena je prikazana na slici 10. Na slici je svaki korak označenom zasebnom bojom.



Slika 10. Maksimalno sažimanje sa korakom 2

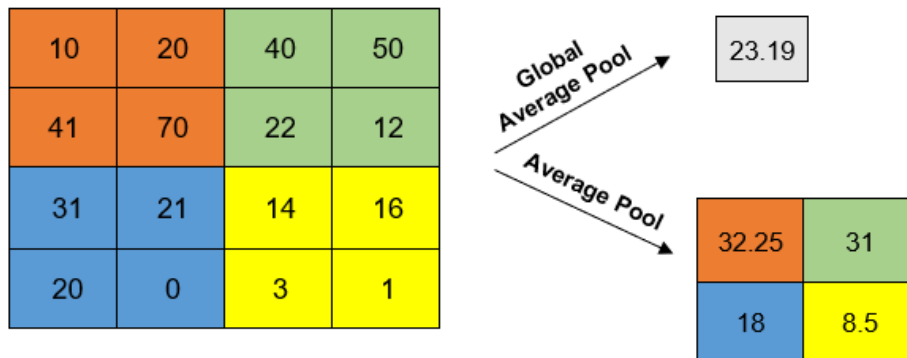
Također se koristi globalno maksimalno sažimanje (eng. *global max pooling*) čime se radi puno veće smanjenje uzorka tako da se uzima samo najveća vrijednost iz čitave mape značajki, slika 11. Ova operacija obično se radi samo jednom prije potpuno povezanog sloja. Globalno maksimalno sažimanje značajno smanjuje broj parametara koje mreža mora optimirati i omogućuje konvolucijskoj neuronskoj mreži da prihvaća ulaze različitih veličina.



Slika 11. Globalno maksimalno sažimanje

Postoji i globalno prosječno sažimanje (eng. *global average pooling*) koje uzima prosječnu vrijednost cijele mape značajki i prosječno sažimanje (eng.

average pooling) koje uzima prosječnu vrijednost dijela mape značajki. Na slici 12 prikazani su primjeri operacija globalnog prosječnog sažimanja i prosječnog sažimanja.



Slika 12. Globalno prosječno i prosječno sažimanje

3.1.3 Sloj izravnavanja

Nakon sloja sažimanja dolazi sloj izravnavanja (eng. *flatten layer*) koji kao ulazni parametar prima mapu značajki. Njegova funkcija je da spljošti mapu značajki, odnosno da pretvori dvodimenzionalnu mapu značajki u jednodimenzionalni vektor kako bi te podatke mogli proslijediti dalje kroz mrežu, obično u potpuno povezani sloj. Na slici 13 prikazana je operacija izravnavanja (eng. *flattening*) s kojom dobivamo linearni vektor koji služi kao ulaz za daljnje slojeve neuronske mreže.

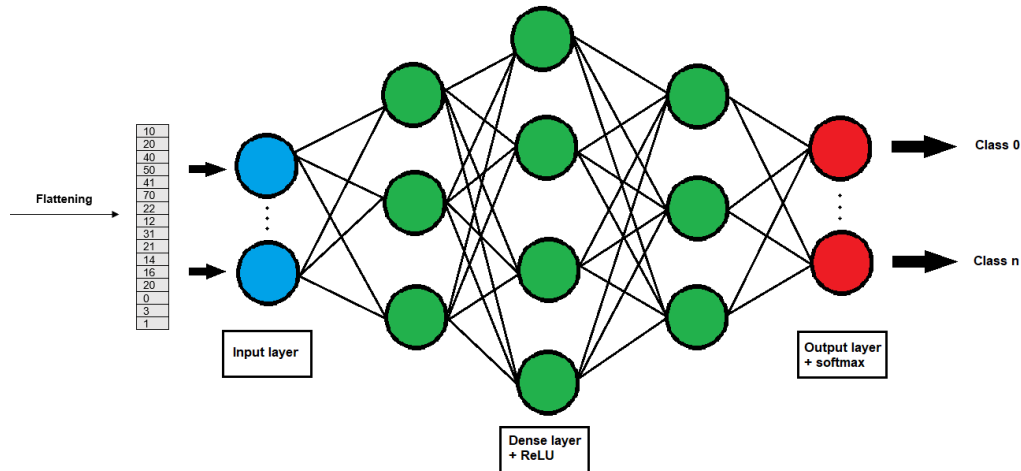


Slika 13. Operacija izravnavanja

3.1.4 Potpuno povezani sloj

Konačna mapa značajki, odnosno mapa značajki posljednjeg konvolucijskog sloja ili posljednji izlaz sloja sažimanja koji se pretvara u jednodimenzionalni vektor operacijom izravnavanja, povezuje se na jedan ili više potpuno povezanih slojeva. Potpuno povezani sloj koji se inače naziva gusti sloj (eng. *dense layer*) je sloj neuronske mreže u kojem je svaki ulaz povezan na svaki izlaz s težinom koju je moguće optimirati tijekom procesa učenja (Yamashita, 2018). Na izlazu, odnosno posljednjem sloju mreže, obično se nalaze vjerojatnosti svake klase koje rješava taj klasifikacijski problem. Svaki potpuno povezani sloj također sadrži neku nelinearnu funkciju, najčešće *ReLU*. Aktivacijska funkcija koja se primjenjuje na zadnji potpuno povezani sloj obično se razlikuje od ostalih i ovisi o problemu kojeg je potrebno riješiti. Ako je u pitanju višeklasni problem najbolje je koristiti *softmax* aktivacijsku funkciju jer ona normalizira vrijednosti iz zadnjeg sloja pa na izlazu dobivamo vrijednosti u rasponu od nula do jedan čiji je zbroj jednak 1. Za binarnu klasifikaciju koristi se sigmoida, a u slučaju regresije gdje je cilj dobiti neku kontinuiranu vrijednost koristi se funkcija identiteta (linearna funkcija). Na slici 14 prikazani su potpuno povezani slojevi koji kao ulaz primaju vektor, a kao

aktivacijsku funkciju koriste *ReLU*. Zadnji sloj koristi funkciju *softmax*, jer je riječ o problemu višeklasne klasifikacije.



Slika 14. Potpuno povezani sloj

3.2 Učenje konvolucijske neuronske mreže

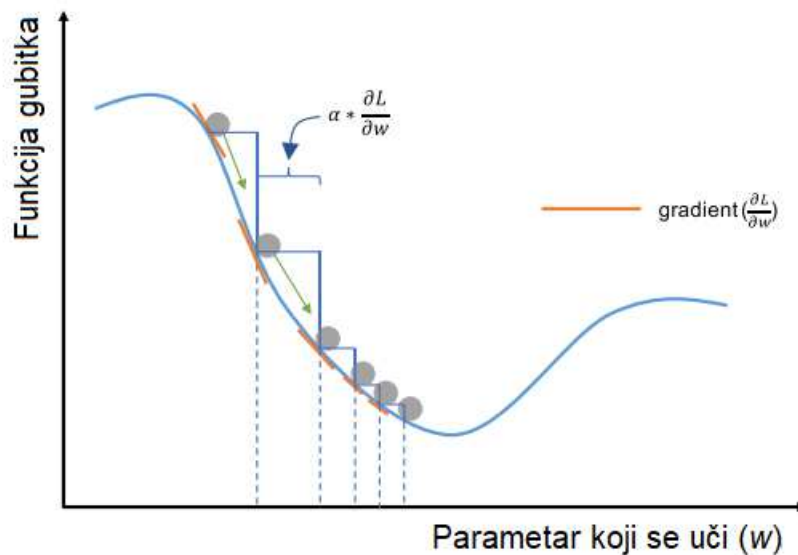
Proces učenja mreže svodi se na pronalazak optimalnih detektora značajki u konvolucijskim slojevima i težina u potpuno povezanim slojevima. Cilj je optimirati navedene parametre tako da se minimizira razlika predikcija koje daje model i danih istinitih predikcija na skupu za učenje. Algoritam unazadne propagacije (eng. *backpropagation*) obično se koristi za učenje neuronske mreže.

Učinkovitost modela s određenim vrijednostima parametara ocjenjuje se na temelju rezultata funkcije gubitka u unaprijednoj propagaciji (eng. *forward propagation*) na skupu za učenje. Uspoređuju se predikcije modela sa stvarnim vrijednostima (Yamashita, 2018). Za višeklasnu klasifikaciju kao funkcija gubitka najčešće se koristi unakrsna entropija (eng. *cross entropy*), a za regresiju se najčešće koristi srednja kvadratna pogreška (eng. *mean square error*).

Nakon toga, težine i detektori značajki ažuriraju se pomoću vrijednosti funkcije gubitka u algoritmu unazadne propagacije i primjene gradijentnog spusta. Gradijentni spust iterativno optimira parametre mreže kako bi minimizirao funkciju gubitka. Gradijent funkcije gubitka daje smjer najveće stope porasta funkcije pa se stoga svaki parametar funkcije ažurira u suprotnom smjeru gradijenta pomnoženog s proizvoljnom vrijednosti hiperparametra zvanim stopa učenja kao što je prikazano na slici 15 (Yamashita, 2018). Prema formuli 4, gradijent je parcijalna derivacija funkcije gubitka L po parametru w pomnožena sa stopom učenja α koju je kao hiperparametar potrebno specificirati prije pokretanje procesa učenja mreže.

$$w := w - \alpha * \frac{\partial L}{\partial w}$$

Formula 4.



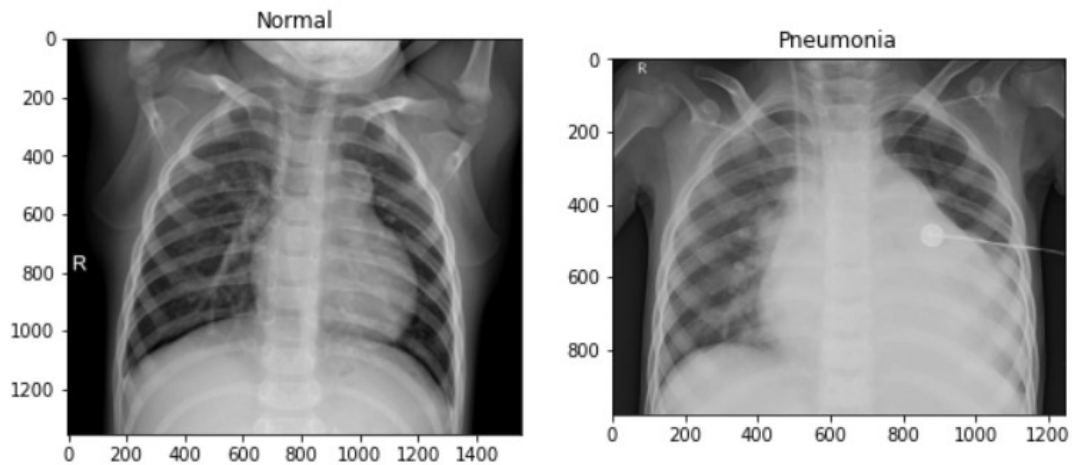
Slika 15. Gradijentni spust (Yamashita, 2018)

S obzirom na to da postoje memorijska ograničenja i često ne možemo obraditi sve podatke odjednom kod učenje mreže najčešće se koristi

stohastički gradijentni spust (eng. *stochastic gradient descent* – SGD). U SGD-u gradijent funkcije gubitka uz određene parametre mreže računa se koristeći podskup skupa za učenje koji nazivamo *mini-batch* i čija vrijednost se specificira prije početka učenja (Yamashita, 2018). Nakon obrade svakog podskupa računa se funkcija gubitka i ažuriraju se parametri. Postoje i brojne druge optimizacije gradijentnog spusta kao što su *Adam*, *RMSprop* itd.

4. Skup podataka

U ovom diplomskom radu razvijena je konvolucijska mreža za dijagnosticiranje upale pluća (pneumonije). Podaci koji su korišteni za učenje i testiranje modela preuzeti su sa stranice *kaggle.com* i sastoje se od rendgenskih snimaka prsnog koša. Sve rendgenske snimke prsnog koša prikupljene su iz medicinskog centra za žene i djecu, Guangzhou. Sve snimke rađene su kao dio pacijentova rutinskog kliničkog pregleda. Sve rendgenske snimke pregledane su radi kontrole kvalitete i uklonjene su sve snimke niske kvalitete. Dijagnoza svake snimke donesena je od strane dvaju stručnjaka prije nego je dozvoljena njihova uporaba za učenje sustava umjetne inteligencije. Kako bi se uzele u obzir moguće pogreške kod dijagnosticiranja, evaluacijski skup provjerio je i treći stručnjak. Na slici 16 prikazani su primjeri slika obje kategorije, tj. zdravog i oboljelog pacijenta. Na snimci pacijenta koji boluje od upale pluća vide se zamagljenja na plućima koja pokazuju da pacijent vjerojatno boluje od upale pluća.

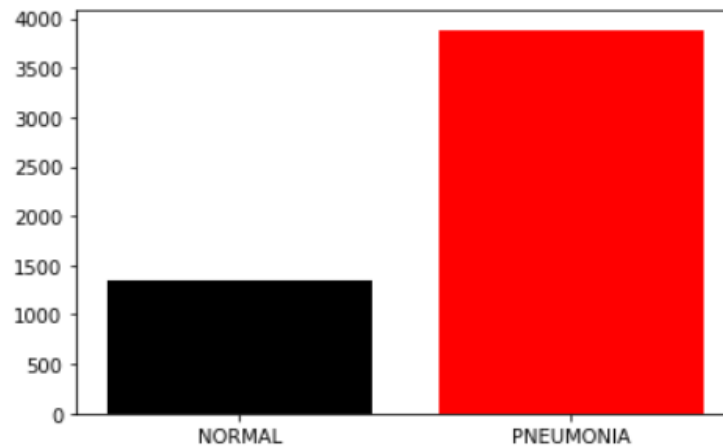


Slika 16. Pacijent bez i s upalom pluća

Skup podataka koji se koristi za učenje i testiranje modela ima više od pet tisuća slika. Distribucija slika po klasama za svaki skup, napravljena od strane autora, prikazana je na slici 2 i 3. Iz skupa za učenje također je odvojeno dvadeset posto slika koje služe kao validacijski skup.

Train data distribution:

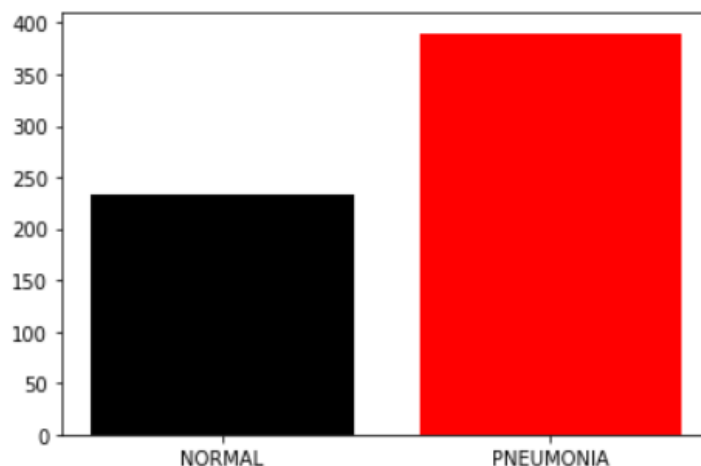
`{'NORMAL': 1350, 'PNEUMONIA': 3884}`



Slika 17. Distribucija skupa za učenje

Test_data_distribution:

```
{'NORMAL': 234, 'PNEUMONIA': 390}
```



Slika 18. Distribucija skupa za testiranje

5. Tehnike evaluacije performansi modela

Rezultati naučenih neuronskih mreža u ovom diplomskom radu bit će evaluirani korištenjem nekoliko mjera performansi modela. U idućim formulama klasificiran pozitivan primjerak (eng. *true positive*, TP) predstavlja pacijenta koji boluje od upale pluća i model je to ispravno procijenio, a negativno klasificiran primjerak predstavlja pacijenta koji ne boluje od upale pluća (eng. *true negative*, TN). Pogrešne klasifikacije modela predstavljaju lažno istinit slučaj (eng. *false positive*, FP), tj. snimke rendgena koje je model pogrešno klasificirao kao pacijenta s upalom pluća i lažno negativan slučaj (eng. *false negative*, FN) gdje je model klasificirao snimku tj. pacijenta kao zdravog, a zapravo ima upalu pluća. Navedena opažanja obično se prikazuju matricom konfuzije (eng. *confusion matrix*) koja daje uvid u rezultate

klasifikacijskog modela na danom testnom skupu u kojem su poznate istinite vrijednosti. Na slici 19 prikazan je primjer matrice konfuzije.

N=165	Predicted = NO	Predicted = YES	
Actual = NO	TN = 50	FP = 10	60
Actual = YES	FN = 5	TP = 100	105
	55	110	

Slika 19. Matrica konfuzije

Točnost (eng. *accuracy*) je ukupan broj točno klasificiranih snimaka podijeljen s ukupnim brojem snimaka. Točnost je odlična mjera samo kada imamo simetrični skup podataka gdje su vrijednosti lažno pozitivnih i lažno negativnih skoro podjednake. U suprotnom je potrebno uzeti i druge mjere u obzir kako bi se dobio bolji uvid u performanse modela.

$$Accuracy = \frac{(TP + TN)}{(TP + FN) + (FP + TN)}$$

Formula 5.

Odziv (eng. *recall, sensitivity*) predstavlja omjer ispravno predviđenih pozitivnih opažanja podijeljenih sa svim opažanjima koja su zapravo pozitivna, odnosno postotak točno klasificiranih pacijenata s upalom pluća u odnosu na ukupan broj pacijenata s upalom pluća u danom skupu.

$$Sensitivity = \frac{(TP)}{(TP + FN)}$$

Formula 6.

Preciznost (eng. *precision*) predstavlja omjer ispravno predviđenih pozitivnih opažanja podijeljenih s ukupnim brojem predviđenih pozitivnih opažanja, odnosno postotak u kojem je model u pravu kada predvidi upalu pluća.

$$Precision = \frac{(TP)}{(TN + FP)}$$

Formula 7.

Mjera *F1* (eng. *F1-score*) kombinira odziv i preciznost klasifikatora u jednu mjeru na način da uzima njihovu harmonijsku sredinu. Na ovaj način u obzir se uzimaju lažno pozitivna i lažno negativna opažanja. Mjera *F1* daje puno bolju evaluaciju performansi modela kada skup podataka koji se obrađuje ima nejednaku distribuciju klasa.

$$F1 - Score = \frac{2 * Sensitivity * Precision}{Sensitivity + Precision}$$

Formula 8.

6. Implementacija konvolucijske neuronske mreže

Za implementaciju se koristi programski jezik Python, a za izradu konvolucijske neuronske mreže korištena je biblioteka *Keras*. U diplomskom radu se za razvojnu okolinu koristi *Google Colab* koji omogućuje jednostavno korištenje GPU-a za učenje modela što značajno ubrzava proces. Velik broj jezgara grafičke kartice omogućava znatno brže izračunavanje paralelnih procesa kojih ima jako puno s obzirom na veliku količinu parametara koji se optimiraju tijekom procesa učenja mreže.

6.1 Predobrada skupa podataka

Ulazni podaci, u ovom slučaju rendgenske snimke prsnog koša, moraju se obraditi i pretvoriti u tenzore s vrijednostima s pomičnim zarezom (*float*) kako bi ih konvolucijska neuronska mreža mogla obraditi. Snimke su spremljene u formatu *jpg* različitih dimenzija pa su sve svedene na istu dimenziju $350 \times 350 \times 1$, gdje 1 označava broj kanala što u ovom slučaju predstavlja crno-bijele slike. *Batch size* predstavlja broj primjeraka koji se obrađuje u jednoj iteraciji tijekom procesa učenja, a u ovom slučaju je to 32. Za postupak predobrade podataka koristi se klasa *ImageDataGenerator* biblioteke *Keras*. Pomoću navedene klase moguće je jednostavno umjetno povećati broj skupa podataka za učenje kako bi se povećala raznolikost i u konačnici poboljšala točnost predikcija modela na neviđenim snimkama. Svaka digitalna slika sastoji se od piksela u rasponu od 0 do 255. Podešavanjem parametra *rescale* klase *ImageDataGenerator* na vrijednost $1./255$ dobiven je vektor s rasponom brojeva između 0 i 1 koji omogućuje da slike ravnomjernije doprinose funkciji gubitka. Veći raspon piksela zahtijeva manju stopu učenja zbog veće vrijednosti funkcije gubitka, a manji raspon zahtijeva veću stopu učenja. Parametar raspon zumiranja (eng. *zoom range*) omogućuje povećanje slike ako je vrijednost manja od 1, a u suprotnom se dodatno udalji

od slike. Parametri horizontalno i vertikalno okretanje (*eng. horizontal and vertical flip*) omogućuju nasumično okretanje slike horizontalno ili vertikalno. Korišten je i parametar raspon smicanja (*eng. shear range*) koji predstavlja transformaciju smicanja, odnosno fiksiranje jedne osi i rastezanje slike pod određenim kutom koji se naziva kut smicanja. Sve ove transformacije nasumično se primjenjuju na slike, osim skaliranja koje je primijenjeno na sve slike. Skup za testiranje samo je skaliran kako bi se izbjegla manipulacija podataka prije testiranja. Na slici 20 prikazan je dio koda koji prikazuje predobradu skupa podataka i učitavanje slika. Na skupu za testiranje primijenjeni su horizontalno okretanje (*eng. horizontal flip*), raspon zumiranja (*eng. zoom range*), raspon smicanja (*eng. shear range*). Validacijski skup, skup za testiranje i skup za učenje su svi skalirani tako da im vrijednosti budu u rasponu od 0 do 1. Za učitavanje se koristi metoda *flow from directory* koja dohvaća sve slike iz navedenih mapa. Metoda kao prvi parametar prima put do direktorija gdje se nalaze skup za testiranje i učenje mreže. Skup za validaciju stvara se tako da uzmemo dvadeset posto slika iz skupa za učenje. Parametar *target size* predstavlja veličinu ulaznih slika. *Color mode* je podešen na *grayscale* jer su rendgenske snimke crno bijele. Parametar *class mode* je podešen na binarno jer postoje samo dvije klase za predviđanje.

Generate Data for model

```
▶ batch_size = 32
   target_size = (350, 350)

   train_data_gen = ImageDataGenerator(rescale= 1./255,
                                       validation_split =0.2,
                                       zoom_range=0.55,
                                       shear_range = 0.45,
                                       horizontal_flip=True,
                                       )

   test_data_gen = ImageDataGenerator(rescale = 1./255)

   train_generator = train_data_gen.flow_from_directory(train_folder,
                                                       target_size=target_size,
                                                       batch_size=batch_size,
                                                       subset='training',
                                                       color_mode='grayscale',
                                                       class_mode='binary')

   val_generator = train_data_gen.flow_from_directory(train_folder,
                                                    target_size=target_size,
                                                    batch_size=batch_size,
                                                    subset='validation',
                                                    color_mode='grayscale',
                                                    class_mode='binary')

   test_generator = test_data_gen.flow_from_directory(test_folder,
                                                    target_size=target_size,
                                                    batch_size = batch_size,
                                                    class_mode = 'binary',
                                                    color_mode='grayscale',
                                                    shuffle=False)

↳ Found 4187 images belonging to 2 classes.
   Found 1045 images belonging to 2 classes.
   Found 624 images belonging to 2 classes.
```

Slika 20. Predobrada i učitavanje podataka

6.2 Model konvolucijske neuronske mreže

Sažetak modela prikazan je na slici 21. Model koji je naučen u ovom diplomskom radu sastoji se od 8 konvolucijskih slojeva, 4 sloja maksimalnog sažimanja (eng. *max pooling layer*) za smanjenje dimenzionalnosti, 6 slojeva *dropout* i 2 potpuno povezana sloja. Sloj *dropout* predstavlja metodu

regularizacije koja aproksimira učenje velike neuronske mreže s različitim arhitekturama paralelno. Tijekom učenja određeni postotak nasumično odabranih izlaza pojedinih slojeva je ignoriran što sprječava prenaučenos modela. U prvom konvolucijskom sloju operacije se obavljaju na slikama veličine (350, 350) uz veličinu detektora značajki (3, 3), korak 1 i dopunu 'valid', što označava da se ne koristi nikakva metoda dopunjavanja, pa je izlaz veličine $(350 - 3 + 1, 350 - 3 + 1) = (348, 348)$. Ostali slojevi koriste dopunu 'same' koja predstavlja dopunu nulama jednako s lijeve i desne strane ili gore i dolje te uz korak 1 osigurava da su dimenzije ulaza jednake kao dimenzije izlaza. Broj detektora značajki u prva dva sloja je 32 pa je izlaz sada (None, 348, 348, 32) gdje *None* predstavlja dodatnu dimenziju koju *Keras* automatski dodaje jer veličina *batcha* može varirati. Nakon dva konvolucijska sloja slijedi sloj maksimalnog sažimanja s pomakom (2, 2) i veličinom jezgre (2, 2) koja primijenjena na ulaz veličine (348, 348) rezultira izlazom $((348 - 2 // 2) + 1, (348 - 2 // 2) + 1) = (174, 174)$. Aktivacijska funkcija kroz koju prolazi izlaz operacije konvolucije je *ReLU*. Nakon toga slijedi sloj *dropout*. Na isti način složen je i ostatak mreže uz sve veći broj filtara, tj. detektora značajki. Nakon toga slijedi sloj izravnavanja koji uzima sve piksele i kanale i stvara jednodimenzionalni vektor. Iz ulaza veličine (21, 21, 256) stvara vektor od $(21 * 21 * 256) = 112896$ vrijednosti. Nakon toga slijede 2 potpuno povezana sloja veličine 256 i 512 sa slojevima *dropout*. Budući da je riječ o klasifikaciji samo dvije klase, za posljednji sloj kao aktivacijska funkcija koristi se sigmoida koja vraća vrijednosti u rasponu od 0 do 1 (s vrijednosti većom ili jednakom 0.5 za klasu 1, a manje od toga za klasu 0). Model ima ukupno preko 30 milijuna parametara koje je potrebno optimirati procesom učenja.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 348, 348, 32)	320
conv2d_1 (Conv2D)	(None, 348, 348, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 174, 174, 32)	0
dropout (Dropout)	(None, 174, 174, 32)	0
conv2d_2 (Conv2D)	(None, 174, 174, 64)	18496
conv2d_3 (Conv2D)	(None, 174, 174, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 87, 87, 64)	0
dropout_1 (Dropout)	(None, 87, 87, 64)	0
conv2d_4 (Conv2D)	(None, 87, 87, 128)	73856
conv2d_5 (Conv2D)	(None, 87, 87, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 43, 43, 128)	0
dropout_2 (Dropout)	(None, 43, 43, 128)	0
conv2d_6 (Conv2D)	(None, 43, 43, 256)	295168
conv2d_7 (Conv2D)	(None, 43, 43, 256)	590080
max_pooling2d_3 (MaxPooling2D)	(None, 21, 21, 256)	0
dropout_3 (Dropout)	(None, 21, 21, 256)	0
flatten (Flatten)	(None, 112896)	0
dense (Dense)	(None, 256)	28901632
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513

=====
Total params: 30,205,409
Trainable params: 30,205,409
Non-trainable params: 0

Slika 21. Model

6.3 Prevođenje i prilagodba modela

Prije prevođenja modela potrebno je podesiti stopu učenja, funkciju gubitka i metriku. Budući da je riječ o binarnoj klasifikaciji kao funkcija gubitka odabrana je binarna unakrsna entropija (eng. *binary cross entropy*). Za metriku je odabrana točnost (eng. *accuracy*) koja računa koliko često stvarne vrijednosti odgovaraju predikcijama modela. Tijekom procesa učenja mjerit će se gubitak i točnost skupa za učenje i validacijskog skupa. Za stopu učenja početna vrijednost je 0.0001, a za optimiranje stohastičkog gradijentnog spusta odabran je optimizator *Adam*. Ova metoda je učinkovita kada problem koji rješavamo ima puno podataka i parametara zbog svojih manjih memorijskih zahtjeva. Intuitivno, to je kombinacija gradijentnog spusta s momentom i algoritma *RMSP*. Gradijentni spust s momentom se koristi za ubrzanje gradijentnog spusta tako da se u obzir uzima i eksponencijalno otežani prosjek koji ubrzava konvergenciju algoritma prema minimumu. *RSMP (Root Mean Square Propagation)* je adaptivni algoritam učenja koji pokušava poboljšati *AdaGrad*. Umjesto uzimanja kumulativne sume kvadrata gradijenata kao u *AdaGradu* uzima se „eksponencijalni pomični prosjek“ (eng. *exponential moving average*). Optimizator *Adam* nasljeđuje prednost obje metode i nadovezuje se na njih kako bi pružao optimalan gradijentni spust tako da postoje minimalne oscilacije kada se dosegne globalni minimum, a prije toga se rade dovoljno veliki koraci kako bi se prošli lokalni minimumi (GeekForGeeks, 2020). Na slici 22 prikazan je isječak koda koji prikazuje prevođenje (eng. *compile*) modela.

```
[ ] model.compile(loss='binary_crossentropy',  
                 optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),  
                 metrics=['accuracy'],)
```

Slika 22. Prevođenje modela

Nakon toga potrebno je definirati metode prilagodbe modela za vrijeme učenja. Metoda ranog zaustavljanja (eng. *early stopping*) zaustavlja učenje na određenoj epohi ovisno o zadanim metrikama i uvjetima koji se prate. Na ovaj način sprječavamo prenaučenosť modela. U ovom slučaju prati se vrijednost funkcije gubitka na validacijskom skupu koja treba biti što manja. Parametar strpljenja (eng. *patience*) omogućuje da se učenje zaustavlja nakon što je postignuta npr. minimalna vrijednost funkcije gubitka, a u narednih nekoliko iteracija vrijednost funkcije se povećava. Parametar *restore best weights* postavljen na *True* osigurava da se nakon prekida težine modela vrate na vrijednosti iz epohe u kojoj su postignuti najbolji rezultati. Bolji rezultat dobiven je kada je faza učenja trajala svih 150 epoha, nego kada se koristilo ranije zaustavljanje. Druga metoda je smanjenje stope učenja kada se metrika prestane poboljšavati. Modeli često imaju koristi od smanjenja stope učenja za neki faktor nakon što učenje stagnira. Obje navedene metode spadaju u metode povratnog praćenja (eng. *callback monitoring method*). Prije početka učenja mreže dobra je praksa postaviti težine pojedinih klasa kako bi se naglasila težina manjinske klase, što omogućuje da model podjednako uči od svih klasa. Na slici 23 prikazana je implementacija metoda ranog zaustavljanja, komentirane prilikom izvođenja, i smanjenje stope učenja te podešavanje težina klasa. Nakon što je sve podešeno, pozivom metode *fit* uz predaju skupa za učenje, validacijskog skupa, metoda povratnog praćenja, težina klasa i broja epoha koje će izvoditi, započinje učenje mreže.

```

with tf.device('/device:GPU:0'):
    #earlystop =tf.keras.callbacks.EarlyStopping(monitor='val_loss',
    #      min_delta=0,
    #      patience=15,
    #      verbose=1,
    #      restore_best_weights=True)
    reduce_lr = tf.keras.callbacks.ReduceLRonPlateau(monitor='val_loss',
    factor=0.5,
    patience=8,
    verbose=1,
    min_delta=0.0001)
    weight_for_0 = (1.0 / 1350)*(5234/2.0)
    weight_for_1 = (1.0 / 3884)*(5234/2.0)
    #print(weight_for_0,weight_for_1)
    class_weight = {0: weight_for_0, 1: weight_for_1}

    callback=[reduce_lr,]

    results = model.fit(train_generator, epochs=150 ,verbose=1,
    validation_data=val_generator,callbacks=callback,class_weight=class_weight,)

```

Slika 23. Prilagodba i učenje modela

6.4 Rezultati naučene neuronske mreže

Nakon 150 epoha model je ostvario točnost od 97.83% na skupu za učenje, 95.21% na validacijskom skupu i 93.27% na testnom skupu, kao što je i prikazano na slici 24.

```

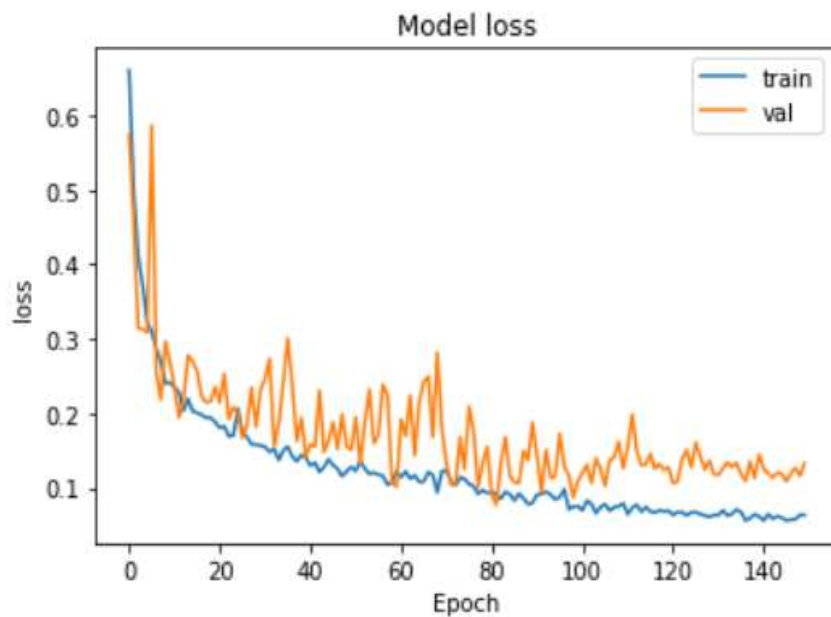
33/33 [=====] - 12s 355ms/step - loss: 0.1330 - accuracy: 0.9522
Val loss: 0.13303957879543304
Val accuracy: 0.9521530866622925
131/131 [=====] - 78s 593ms/step - loss: 0.0637 - accuracy: 0.9783
Train loss: 0.06366149336099625
Train accuracy: 0.9782660603523254
20/20 [=====] - 152s 8s/step - loss: 0.3220 - accuracy: 0.9327
Test loss: 0.32203230261802673
Test accuracy: 0.932692289352417

```

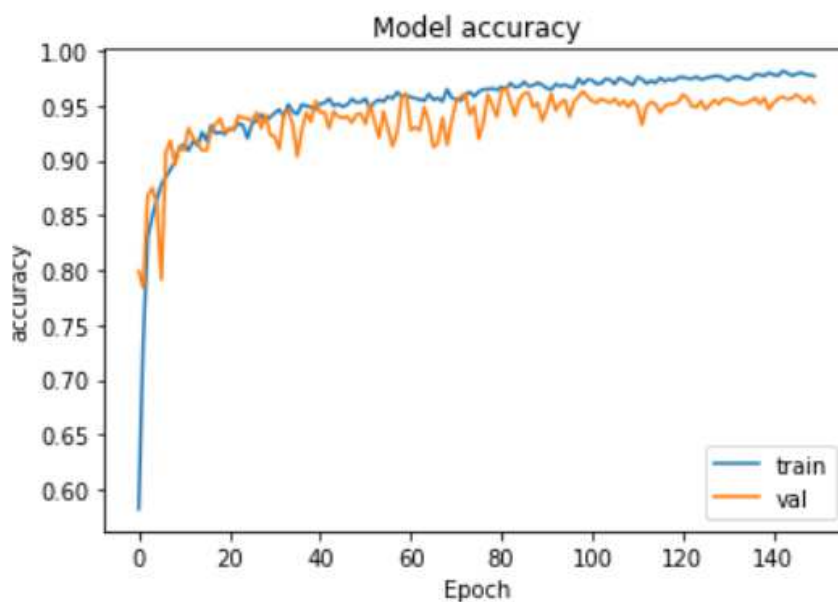
Slika 24. Rezultati modela

Na iduća dva grafa prikazan je napredak modela za vrijeme učenje. Iz slika 25 i 26 možemo vidjeti da nakon 100 epoha točnost polako počinje stagnirati, a vrijednost funkcije gubitka na validacijskom skupu stabilizirala se oko

vrijednosti 0.133. Daljnje učenje mreže vjerojatno bi dovelo do prenaučnosti modela.

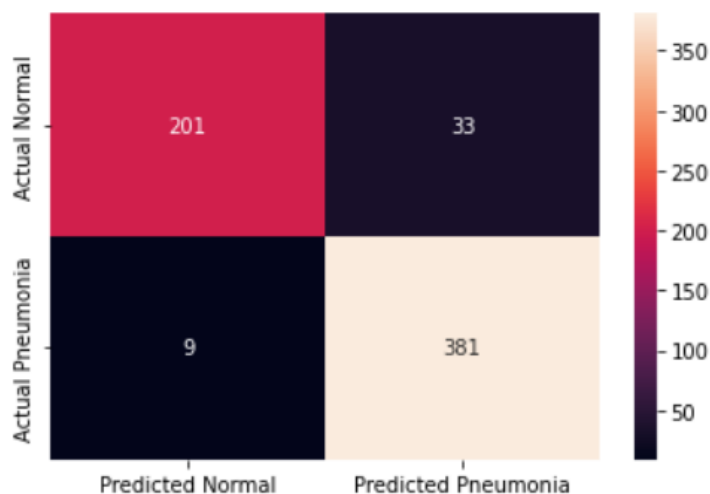


Slika 25. Gubitak modela za vrijeme učenja



Slika 26. Točnost modela za vrijeme učenja

Nakon toga obavljena je predikcija na testnom skupu i na temelju rezultata stvorena je matrica konfuzije prikazana na slici 27.



Slika 27. Matrica konfuzije na testnom skupu

Iz matrice se vidi da model jako dobro klasificira pacijente koji boluju od upale pluća jer od 390 snimaka, gdje je prisutna upala pluća, model je točno klasificirao njih 381. Model također ima više lažno pozitivnih primjera, tj. 33, nego lažno negativnih kojih je samo 9. U području medicinske dijagnostike najbitnije je predvidjeti prisutnost bolesti kako bi se moglo započeti s liječenjem. U rijetkim slučajevima model neće predvidjeti da pacijent boluje od upale pluća. Nešto veći broj lažno detektirane upale pluća nije idealan, ali je prihvatljivije imati nešto više lažno detektiranih slučajeva nego propustiti dijagnosticirati upalu pluća, što može uzrokovati ozbiljne posljedice po pacijenta.

Na temelju matrice konfuzije izračunate su točnost, preciznost, odziv i mjera *F1* na skupu za testiranje. Rezultati su prikazani u tablici 1.

Tablica 1. Izvješće o klasifikaciji

Točnost	Preciznost	Odziv	Mjera <i>F1</i>
0.9327	0.92	0.977	0.947

7. Rezultati i metode iz drugih radova

Rahman i suradnici u svom radu koristili su četiri unaprijed naučene konvolucijske neuronske mreže *AlexNet*, *ResNet18*, *DenseNet201* i *SqueezeNet* za prijenosno učenje (eng. *transfer learning*) kako bi detektirali upalu pluća s rendgenskih snimaka. Korišten je isti skup podataka kao i u ovom diplomskom radu, a skup podataka podijeljen je na skup za učenje i skup na testiranje. Prijenosno učenje je metoda strojnog učenja gdje se model, razvijen za neki problem, ponovno koristi kao početna točka razvoja novog modela za neki drugi problem. U radu Rahmana i suradnika snimke su prethodno obrađene tako da su im dimenzije prilagođene zahtjevima unaprijed naučenih mreža i sve su slike normalizirane. Budući da skup podataka nije pretjerano velik, autori su primijenili nekoliko tehnika povećanja broja primjera (eng. *data augmentation*) kako bi poboljšali točnost klasifikacije i spriječili prenaučenosť modela. Napravili su rotaciju slika za 315 stupnjeva, operaciju skaliranja tj. povećanja slike za 10 posto i translaciju slika horizontalno i vertikalno za 10 posto. U tablici 1 prikazani su parametri sva četiri unaprijed naučena modela.

Tablica 2. Parametri učenja modela (Rahman, 2020)

CNN model	Veličina slike	Optimizacija	Moment	<i>Mini-Batch</i>	Stopa učenja
AlexNet	227x227	Gradijentni spust	0.9	16	0.0003
ResNet18	224x224				
DenseNet201	224x224				
SqueezeNet	227x227				

Rezultati koje dobili Rahman i suradnici u svom radu prikazani su u tablici 2. Iz tablice je vidljivo da je najvišu točnost i najbolju mjeru $F1$ dobio model *DenseNet201* koji je postigao točnost od 98% (Rahman, 2020).

Tablica 3. Rezultati modela (Rahman, 2020)

Model	Točnost	Odziv	Preciznost	Mjera $F1$
<i>AlexNet</i>	0.945	0.953	0.931	0.943
<i>ResNet18</i>	0.964	0.97	0.954	0.965
<i>DenseNet201</i>	0.98	0.99	0.97	0.981
<i>SqueezeNet</i>	0.961	0.94	0.985	0.961

U radu Stephena i suradnika korišten je isti skup podataka podijeljen tako da je za učenje mreže uzeto 3722 slike i 2134 slike za validaciju. U procesu prethodne obrade i augmentacije slika s ciljem povećanja veličine i kvalitete skupa podataka, kako bi se riješio problem prenaučivosti i poboljšala sposobnost generalizacije modela tijekom učenja, autori su primijenili nekoliko operacija. Slike iz skupa za učenje su nasumično rotirane za 40 stupnjeva, translaticirane za 20 stupnjeva horizontalno i vertikalno, preokrenute vodoravno, zumirane za 20% i operacijom smicanja od 20% kutovi slike su izobličeni u smjeru suprotnom kazaljki na satu. Sve slike su također skalirane da im vrijednosti budu u rasponu od 0 do 1. Model korišten u navedenom radu sastoji se od 4 konvolucijska sloja s veličinom detektora značajki 3x3 i njihovim brojem u rasponu od 32 do 128 te aktivacijskom funkcijom *ReLU* između njih. Iza toga slijedi sloj maksimalnog sažimanja koji se pomoću sloja izravnivanja, na koji se nadovezuje sloj *dropout* pretvara u

jednodimenzionalno polje koje služi kao ulaz u potpuno povezane slojeve neuronske mreže. Mreža se sastoji od dva potpuno povezana sloja veličina 512 i 1, između njih funkcija *ReLU* i na kraju sigmoida koja obavlja klasifikaciju. Za veličinu slike odabrane su dimenzije 200x200x3 i dobivena je točnost od 93.73% na validacijskom skupu. Autori nisu naveli da postoji zasebni testni skup (Stephen, 2019).

U radu Saraiva i suradnika na istom skupu podataka razvijena je konvolucijska neuronska mreža koja na ulazu prima slike veličine 150x150. Sastoji se od 3 konvolucijska sloja s veličinom detektora značajki 3x3 između kojih je sloj maksimalnog sažimanja. Svaki konvolucijski sloj koristi aktivacijsku funkciju *ReLU*. Nakon toga slijedi potpuno povezani sloj koji koristi funkciju *softmax* za predviđanje. Također se koriste i slojevi *dropout*. U ovom radu dobivena je točnost od 94.4%, preciznost 94.3% i odziv 94.5% te mjera *F1* u iznosu od 94.4% (Saraiva, 2019).

Ayan i suradnici u svom radu koristili su isti skup podataka. Za učenje su koristili 5232 slike, a za validaciju i testiranje po 624. Od metoda augmentacije koristili su zumiranje, rotaciju za 40 stupnjeva, okretanje i pomicanje. U svom radu koristili su metodu prijenosnog učenja tako da su fiksirali neke od ranijih slojeva unaprijed naučenih mreža i učili su model isključujući te slojeve. Koristili su dvije CNN mreže, *Xception* i *VGG16*. *Xception* model ima 36 konvolucijskih slojeva iza kojih slijedi sloj logističke regresije. Na mreži su fiksirali prvih deset slojeva, a ostale su učili na skupu za učenje. Također su dodali dva potpuno povezana sloja veličine 1024 i 512 te dvosmjerni izlazni sloj sa aktivacijskom funkcijom *softmax*. Za podešen model *Xception* koristili su slike dimenzija 299x299x3. Ostvarili su točnost od 82%, odziv 93.5%, preciznost od 81.6% i mjeru *F1* od 87.15%. U drugom modelu koristili su model *VGG16* koji se sastoji od 16 konvolucijskih slojeva, 144 milijuna parametara, 5 slojeva maksimalnog sažimanja i 3 potpuno povezana sloja te posljednjim slojem s aktivacijskom funkcijom *softmax*. U

ovom slučaju fiksirali su prvih 8 slojeva i učili ostatak. Dobiveni rezultati bili su: točnost 87%, odziv 89.2%, preciznost 91.3% i mjera $F1$ 90% (Ayan, 2019).

Sharma i suradnici u svom radu također su koristili skup podataka od 5863 rendgenske snimke prsnog koša. Podijelili su skup u 3 podskupa. Za učenje su koristili 5126 slika, za validaciju 191 sliku, a za testiranje 449 slika. Svim slikama promijenjena je veličina na 64x64 piksela, a za augmentaciju je korišteno skaliranje na raspon 0 do 1, rotacija od 20 stupnjeva, zumiranje 20%, horizontalno okretanje, smicanje 20% i pomicanje horizontalno ili vertikalno za 20%. Model s kojim su dobili najbolje rezultate imao je 4 konvolucijska sloja, 2 sloja maksimalnog sažimanja i 3 potpuno povezana sloja veličina 256, 512 i 1. Uz potpuno povezane slojeve je sloj *dropout* koji odbacuje 50% neurona u sloju, a nakon svakog sloja maksimalnog sažimanja je sloj *dropout* koji ignorira 20% parametara. Svi slojevi koriste aktivacijsku funkciju *ReLU* i sigmoida je na izlaznom sloju. Model je učen u 20 epoha koristeći optimizator Adam sa stopom učenja 0.00001 i veličinom *batcha* 32. Dobivena je točnost od 90.68% (Sharma, 2020).

Gonsalves i Xin Ye u svom radu koristili su 5216 slika za učenje konvolucijske mreže i 624 slike za testiranje na istom skupu podataka. Za augmentaciju su koristili skaliranje vrijednosti piksela na raspon od 0 do 1, zumiranje, rotaciju za 45 stupnjeva i horizontalno okretanje. Za optimizaciju gradijentnog spusta kod učenja mreže koristili su optimizator *Adam*. Mreža prihvaća slike veličine 148x148, sastoji se od 4 konvolucijska sloja, 4 sloja maksimalnog sažimanja, 2 sloja *dropout*, po jedan iza drugog i trećeg sloja maksimalnog sažimanja. Nakon toga slijedi sloj izravnivanja koji stvara jednodimenzionalni vektor koji se dalje prosljeđuje u 3 potpuno povezana sloja veličine 512, 1024 i 2. Model je učen u 60 epoha i dobivena je točnost od 94.84% na skupu za učenje te 91.35% na testnom skupu (Gonsalves, 2021).

8. Usporedba rezultata s drugim radovima

Tablica 4. Rezultati radova

Autor	Tehnika	Preciznost	Odziv	Mjera <i>F1</i>	Točnost
Rahman i suradnici [3]	<i>Transfer Learning</i> , DenseNet201	0.97	0.99	0.981	0.98
Ayan i suradnici [21]	<i>Transfer Learning</i> , VGG16	0.913	0.892	0.9	0.87
Sharma i suradnici [22]	CNN model	-	-	-	0.907
Stephen i suradnici [20]	CNN model	-	-	-	0.937
Saraiva i suradnici [25]	CNN model	0.943	0.945	0.944	0.944
Gonsalves i Xin Ye [2]	CNN model	-	-	-	0.914
Ovaj rad	CNN model	0.92	0.977	0.947	0.933

Iz tablice 4 vidljivo je da najbolje rezultate ima konvolucijska mreža *DenseNet201* koja je postigla točnost od čak 98%. S obzirom na to da je riječ o modelu koji je unaprijed učen na znatno većem skupu drugih slika prije nego što je prilagođen i učen na skupu rendgenskih snimaka možemo zaključiti da bi se točnost modela u našem radu mogla povećati kada bi na raspolaganju bio znatno veći skup rendgenskih snimaka ili ako bi se koristilo prijenosno učenje. Također se može primijetiti da model *DenseNet201* ima manje lažno negativnih od lažno pozitivnih predikcija, odnosno broj pacijenata oboljelih od upale pluća koje je model klasificirao kao zdrave bio je manji od broja pacijenata koji su zdravi, a model ih je klasificirao kao oboljele. Isto se može primijetiti i na rezultatima mreže iz ovog rada što predstavlja idealan slučaj u medicinskoj dijagnostici.

9. Zaključak

Upala pluća jedan je od vodećih uzroka smrti u svijetu. Uzrokuje visoki broj bolničkih prijema odraslih osoba uz dosta visoku stopu smrtnosti. Upala pluća može se spriječiti, prema Svjetskoj zdravstvenoj organizaciji (eng. *World Health Organization – WHO*), jednostavnom i ranom dijagnozom i liječenjem. Unatoč tome, velika većina svjetskog stanovništva nema pristup radiološkoj dijagnostici. U prisutnosti tehnologije za snimanje također postoji manjak stručnjaka koji mogu analizirati dobivene rendgenske snimke.

U ovom diplomskom radu predložena je tehnika dubokog učenja za automatsku detekciju upale pluća korištenjem rendgenskih snimaka. Razvijena je konvolucijska neuronska mreža koja je ostvarila točnost od 93.3%, preciznost 92%, odziv 97.7% te mjeru *F1* 94.7%.

S obzirom na dobivene rezultate i rezultate sličnih radova može se zaključiti da se predložena metoda dijagnosticiranja upale pluća može koristiti samo kao pomoćni alat za donošenje kliničkih odluka. Ispostavilo se da oskudica dostupnih podataka predstavlja ograničenje za ovaj pristup. Modeli dubokog učenja zahtijevaju tisuće slika kako bi mogli razviti što bolju sposobnost generalizacije. Veći skup podataka vjerojatno bi poboljšao performanse modela. Također je važno spomenuti da se rezultati modela dubokog učenja ne mogu jasno objasniti. Za dijagnozu upale pluća potrebno je razumijevanje vidljivih znakova bolesti na rendgenskim snimkama prsnog koša. Jedan od glavnih problema modela dubokog učenja je pravilno i jasno objašnjenje dobivenih rezultata što otežava njihovu uporabu u medicinskoj dijagnostici. Primjena dubokog učenja još uvijek nije postala svakodnevica u medicinskoj dijagnostici, ali njezin daljnji razvoj i istraživanja zasigurno će osigurati značajan doprinos u skoroj budućnosti.

10. Literatura

- [1] B. Normandin, 'Everything you need to know about Pneumonia', studeni 2021., <https://www.healthline.com/health/pneumonia#what-is-pneumonia>, zadnji pristup 22. svibnja 2022.
- [2] S. Gonsalves, Y. Xin, 'Using Neural Networks and Tensorflow to detect the presence of Pneumonia in a Patient', SIGNATURE 4: 28, California State University, 2021.
- [3] T. Rahman, M. E. H. Chowdhury, A. Khandakar, K. R. Islam, K. F. Islam, Z. B. Mahbub, M. A. Kadir, S. Kashem, 'Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray', Applied Sciences 10, br. 9: 3233, 2020.
- [4] I. Rudan, L. Tomaskovic, C. Boschi-Pinto, H. Campbell, 'Global estimate of the incidence of clinical pneumonia among children under five years of age', Bulletin of the World Health Organization, br. 12, str. 895-903, 2004.
- [5] M. I. Neuman, E. Y. Lee, S. Bixby, S. Diperna, J. Hellinger, R. Markowitz, S. Servaes, M. C. Monuteaux, S. S. Shah, 'Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children', Journal of Hospital Medicine, br. 7, str 294-298, travanj 2012.
- [6] S. Albawi, T. A. Mohammed, S. Al-Zawi, 'Understanding of a convolutional neural network', 2017 International Conference on Engineering and Technology (ICET), 2017.
- [7] IBM Cloud Education , 'Deep learning', svibanj 2020., <https://www.ibm.com/cloud/learn/deep-learning>, zadnji pristup 22. svibnja 2022.
- [8] MathWorks, 'Deep learning', <https://uk.mathworks.com/discovery/deep-learning.html>, zadnji pristup 22. svibnja 2022.

- [9] R. Yamashita, M. Nishio, R. K. G. Do, K. Togashi, 'Convolutional neural networks: an overview and application in radiology', Insights into Imaging 9, str. 611–629, 2018.
- [10] SuperDataScience Team, 'Convolutional Neural Networks (CNN): Step 3 – Flattening', kolovoz 2018.,
<https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>, zadnji pristup 22. svibnja 2022.
- [11] SuperDataScience Team, 'Convolutional Neural Networks (CNN): Softmax & Cross-Entropy', kolovoz 2018.,
<https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-softmax-crossentropy>, zadnji pristup 22. svibnja 2022.
- [12] Exsilio Solutions, 'Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures', rujan 2016,
<https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures>, zadnji pristup 22. svibnja 2022.
- [13] Machine Learning Crash Course, 'Classification: Precision and Recall',
<https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>, zadnji pristup 22. svibnja 2022.
- [14] Data school, 'Simple guide to confusion matrix terminology', ožujak 2014., <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology>, zadnji pristup 22. svibnja 2022.
- [15] H. Deshmukh, 'Medical X-ray Image Classification using Convolutional Neural Network', studeni 2020.,
<https://towardsdatascience.com/medical-x-ray-%EF%B8%8F-image-classification-using-convolutional-neural-network-9a6d33b1c2a>, zadnji pristup 22. svibnja 2022.
- [16] J. Brownlee, 'A Gentle Introduction to Dropout for Regularizing Deep Neural Networks', prosinac 2018.,

- <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>, zadnji pristup 22. svibnja 2022.
- [17]** GeeksforGeeks, 'Intuition of Adam Optimizer', listopad 2020., <https://www.geeksforgeeks.org/intuition-of-adam-optimizer/>, zadnji pristup 22. svibnja 2022.
- [18]** J. Brownlee, 'A Gentle Introduction to Transfer Learning for Deep Learning', prosinac 2017., <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>, zadnji pristup 22. svibnja 2022.
- [19]** S. Rajaraman, S. Candemir, I. Kim, G. Thoma, S. Antani, 'Visualization and Interpretation of Convolutional Neural Network Predictions in Detecting Pneumonia in Pediatric Chest Radiographs', Applied Sciences 8, br. 10: 1715, 2018.
- [20]** O. Stephen, M. Sain, U. J. Maduh, D. U. Jeong, 'An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare', Journal of healthcare engineering, 2019.
- [21]** E. Ayan, H. M. Ünver, 'Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning', Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019.
- [22]** H. Sharma, J. S. Jain, P. Bansal, S. Gupta, 'Feature Extraction and Classification of Chest X-Ray Images Using CNN to Detect Pneumonia', 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), str. 227-231, 2020.
- [23]** P. Mooney, 'Chest X-Ray Images (Pneumonia)', ožujak 2018., <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia?resource=download>, zadnji pristup 22. svibnja 2022.
- [24]** World Health Organization, 'Pneumonia', studeni 2021., <https://www.who.int/news-room/fact-sheets/detail/pneumonia>, zadnji pristup 25. svibnja 2022.

- [25] A. A. Saraiva, D. B. S. Santos, N. J. C. Costa, J. V. M. Sousa, N. M. F. Ferreira, A. Valente, S. Soares, 'Models of Learning to Classify X-ray Images for the Detection of Pneumonia using Neural Networks', In Bioimaging, str. 76-83, veljača 2019.

Sažetak

Ovaj rad obrađuje detekciju upale pluća na rendgenskim snimkama uporabom metoda dubokog učenja. Upala pluća je respiratorna bolest koju uzrokuju bakterije ili virusi. Analiza rendgenskih snimaka prsnog koša najčešći je način dijagnosticiranja upale pluća. Razvijena je konvolucijska neuronska mreža koja omogućuje automatsku detekciju upale pluća na rendgenskim snimkama prsnog koša. Objasnjena je svrha i funkcionalnost pojedinih slojeva konvolucijskih mreža te su dobiveni rezultati uspoređeni s rezultatima drugih nedavnih radova. Dobivena je točnost od 93.3% na skupu za testiranje. Predložena metoda mogla bi se koristiti samo kao pomoćni alat za donošenje kliničkih odluka. Konačna odluka ipak treba biti prepuštena stručnoj osobi zbog nedostatka jasnog i pravilnog objašnjenja rezultata koje je predvidio model.

Ključne riječi:

- upala pluća, rendgenska snimka, duboko učenje, konvolucijska neuronska mreža, implementacija, evaluacija

Summary

The subject of this paper is the detection of pneumonia on chest X-rays using deep learning methods. Pneumonia is a respiratory disease caused by a virus or bacteria. Chest X-ray analysis is the most common way to diagnose pneumonia. A convolutional neural network with automatic detection of pneumonia on chest X-rays has been developed. The purpose and functionality of individual CNN layers is explained, and the obtained results are compared with other recent studies. An accuracy of 93.3% was obtained on the test set. The proposed method could only be used as an assistance tool for clinical decision making. However, the final decision should be left to the expert due to the lack of a proper and clear explanation of the results predicted by the model.

Keywords:

- pneumonia, chest X-ray, deep learning, convolutional neural network, implementation, evaluation