

Raspoznavanje bolesti lista biljke uporabom strojnog učenja na temelju slike

Radočaj, Petra

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:425249>

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

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni studij

**RASPOZNAVANJE BOLESTI LISTA BILJKE UPORABOM
STROJNOG UČENJA NA TEMELJU SLIKE**

Završni rad

Petra Radočaj

Osijek, 2020.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 03.09.2020.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na
preddiplomskom sveučilišnom studiju**

Ime i prezime studenta:	Petra Radočaj
Studij, smjer:	Prediplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4126, 18.09.2019.
OIB studenta:	96083571167
Mentor:	Doc.dr.sc. Zdravko Krpić
Sumentor:	Dr.sc. Bruno Zorić
Sumentor iz tvrtke:	
Naslov završnog rada:	Raspoznavanje bolesti lista biljke uporabom strojnog učenja na temelju slike
Znanstvena grana rada:	Umjetna inteligencija (zn. polje računarstvo)
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	03.09.2020.
Datum potvrde ocjene Odbora:	09.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 13.09.2020.

Ime i prezime studenta:

Petra Radočaj

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4126, 18.09.2019.

Turnitin podudaranje [%]:

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Raspoznavanje bolesti lista biljke uporabom strojnog učenja na temelju slike**

izrađen pod vodstvom mentora Doc.dr.sc. Zdravko Krpić

i sumentora Dr.sc. Bruno Zorić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD.....	1
1.1. Zadatak završnog rada	1
2. RASPOZNAVANJE BOLESTI LISTA BILJKE NA TEMELJU SLIKE	2
2.1. Bolesti lista biljke.....	2
2.1.1. Dinamika epidemije.....	3
2.1.2. Patogeni	3
2.1.3. Uspostavljanje dijagnoze.....	4
2.2. Primjena raspoznavanja bolesti lista biljke.....	4
2.3. Baze slika biljaka za raspoznavanje bolesti	6
2.4. Postojeća rješenja	8
3. POSTUPCI KLASIFIKACIJE KAO ALAT ZA RASPOZNAVANJE BOLESTI LISTA BILJKE	10
3.1. Prikupljanje podataka i predobrada.....	12
3.1.1. Histogram za slike u boji.....	13
3.1.2. Segmentacija slika algoritmom k-najbližih susjeda (k-NN).....	14
3.2. Klasifikatori	15
3.2.1. Neuronska mreža.....	15
3.2.2. Konvolucijska neuronska mreža – CNN (engl. <i>Convolutional neural network</i>)...	18
3.2.2.1. Konvolucija.....	19
3.2.2.2. Konvolucijski sloj.....	20
3.2.2.3. Aktivacijska funkcija – ReLU (engl. <i>Rectified Linear Unit</i>).....	21
3.2.2.4. Swish i Mish aktivacijske funkcije.....	23
3.2.2.5. Sloj sažimanja.....	24
3.2.2.6. Potpuno povezani sloj	25
3.3. Neke popularne mrežne arhitekture	26
3.3.1. Opis GoogLeNet mreže i prikaz arhitekture	26
3.3.2. Opis VGGNet mreže i prikaz arhitekture.....	28
3.4. Postupci analize i usporedba performansi.....	29
3.4.1. Vizualizacija na primjeru <i>saliency map</i>	29
3.4.2. Matrica zabune.....	30

3.4.3. Preciznost, odziv, mjera F1 i točnost	31
4. EKSPERIMENTALNA ANALIZA I USPOREDBA PERFORMANSI KLASIFIKATORA NA PROBLEMU KLASIFIKACIJE BOLESTI LISTA BILJKE	32
4.1. Zahtjevi na sustav	32
4.2. Korištene tehnologije	33
4.2.1. Python i Anaconda.....	33
4.2.2. MATLAB.....	33
4.2.3. TensorFlow i Keras.....	34
4.2.4. NumPy i SciPy	34
4.2.5. OpenCV i Matplotlib	34
4.3. Postavke eksperimenta.....	35
4.4. Provedba eksperimenta	36
4.5. Usporedba rezultata	44
4.6. Osvrt i mogućnosti poboljšanja	44
5. ZAKLJUČAK.....	45
LITERATURA.....	46
SAŽETAK	57
ABSTRACT.....	58
ŽIVOTOPIS.....	59
PRILOZI.....	60

1. UVOD

Kao jedna od najstarijih ljudskih djelatnosti, poljoprivreda je i danas esencijalna djelatnost o kojoj izravno ovisi sve veći udio ljudske populacije [1]. Proizvodnja hrane je, uz klimatske promjene, ugrožena i bolestima biljaka. One mogu imati katastrofalne posljedice za održivost obiteljskih poljoprivrednih gospodarstava koja čine i do 80% poljoprivredne proizvodnje u zemljama u razvoju [2]. Iako se u posljednje vrijeme značajno smanjila učestalost i obujam epidemija, od 20% do 30% proizvoda na godišnjoj razini je nepovratno izgubljeno [3]. Problem učinkovite zaštite biljaka ugrožen je neadekvatnom uporabom pesticida koja često dovodi do dugotrajne otpornosti patogena pa time šteti mehanizmima obrane biljaka od napada patogena. Za bolesti koje nemaju vidljivih simptoma ili ih se uoči prekasno, potreban je inovativan pristup otkrivanju i liječenju. Upotrebom strojnog učenja u preciznoj poljoprivredi dobivaju se pravovremene i točne dijagnoze bolesti biljaka te manji trošak financijskih i drugih resursa [4]. Prvi i ujedno najučestaliji simptomi pojavljuju se na listovima biljke [5], stoga se i u ovom radu detekcija bolesti biljke temelji na obliku i boji lista.

U drugom poglavlju detaljno se opisuje problem raspoznavanja bolesti preko lista biljke te prikazuju već postojeća rješenja. Treće poglavlje opisuje metode predobrade slika i strojnog učenja te se analiziraju performanse. U četvrtom poglavlju se implementiraju pojedini algoritmi za raspoznavanje bolesti i uspoređuju rezultati. U petom poglavlju prikazuju se najbolji rezultati te se na temelju njih donosi zaključak.

1.1. Zadatak završnog rada

U teorijskom dijelu rada potrebno je opisati problem raspoznavanja bolesti lista biljaka na temelju slika. Potrebno je prikazati mogućnosti primjene postupaka strojnog učenja te dati opis glavnih koraka prilikom njihove primjene. U praktičnom dijelu rada usporediti ponašanje i uspješnost različitih postupaka prikazanih u teorijskom dijelu rada na podatkovnom skupu bolesti lista biljke dostupnom u literaturi.

2. RASPOZNAVANJE BOLESTI LISTA BILJKE NA TEMELJU SLIKE

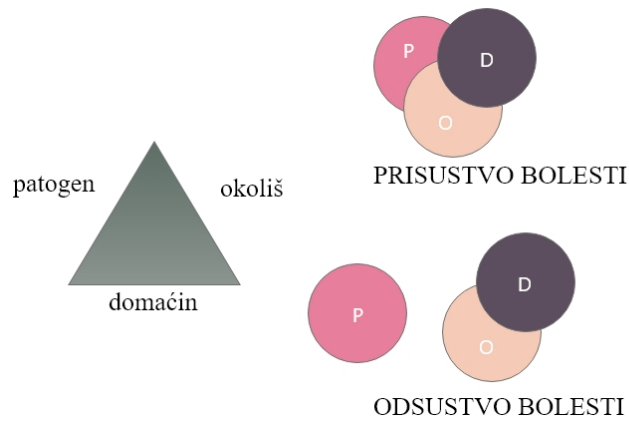
Precizna poljoprivreda podrazumijeva korištenje suvremenih tehnologija pri poljoprivrednoj proizvodnji u svrhu povećanja kvalitete i količine usjeva. Ovakav pristup smanjuje potrebu za manualnom terenskom identifikacijom bolesti. Kartiranje dijelova zemljišta zahvaćenih bolestima, omogućuje primjenu pesticida samo na tom području. To rezultira smanjenjem troškova zaštite usjeva i ima pozitivan učinak na okoliš, u odnosu na uobičajeni pristup pri čemu se pesticidima tretira cijelo zemljište [6].

Raspoznavanje bolesti lista biljke temeljeno na obradi slike omogućava brzo i točno predviđanje razvoja bolesti i kontroliranje njezinog širenja [7]. Primjerice, H. Wang et al. su pomoću modela neuronskih mreža omogućili uspješno raspoznavanje bolesti kineskih žitarica *Puccinia striiformis f. sp. tritici* i *P. recondita f. sp. tritici* koje imaju gotovo jednake simptome u ranom stadiju rasta [8]. Proces raspoznavanja se provodi izdvajanjem karakterističnih značajki bolesnih dijelova biljke predobradom slika i korištenjem odgovarajućeg klasifikatora. Od karakterističnih značajki koje su promatrane, izdvajaju se boja, oblik i tekstura lista [7].

Za bilo koju metodu raspoznavanja objekata potrebna je odgovarajuća baza podataka koja je opisana u ovom poglavlju. Predstavljena su i temeljna znanja o bolestima biljaka potrebna za izradu modela raspoznavanja. Uz opis najčešćih primjena raspoznavanja bolesti lista biljke, opisana su i postojeća rješenja s visokom točnosti raspoznavanja.

2.1. Bolesti lista biljke

Prema [9], bolest biljke definira se kao fiziološki poremećaj ili pojava abnormalne strukture koja je štetna za biljku, njezin dio ili proizvod. Kao rezultat poremećaja, dolazi do smanjenja njezine ekonomske vrijednosti. Pojava simptoma je često prvi indikator bolesti. Najčešće se radi o promjenama zelene boje lista u žutu ili smeđu, šupljikavosti lista ili uvenuću biljke. Iznimka je pojava latentne zaraze, točnije bolesti bez simptoma. Radi se o slučajevima kada nema vidljivih znakova zaraze, ali je uzročnik bolesti prisutan u biljci [9]. Kako bi se bolest razvila, potrebno je zadovoljiti sljedeće uvjete: prisutnost uzročnika bolesti i domaćina te povoljnu okolinu za razvoj bolesti gdje se najčešće promatraju ekološki čimbenici, vlaga i temperatura [9, 10].



Slika 2.1. Trokut pojave biljne bolesti, izrađeno prema [10]

2.1.1. Dinamika epidemije

Masovnoj pojavi bolesti na biljkama pogoduje prisutnost agresivnog uzročnika bolesti, lagano prenošenje, povoljna vlažnost i temperatura te velika reproduktivna sposobnost parazita. Uz prostorno širenje, za svaku epidemiju promatra se i vremenska dinamika širenja bolesti [11]. Mjestimične epidemije mogu rapidno smanjiti kvalitetu usjeva i uništiti ih [12]. Od iznimne je važnosti zato spriječiti eskalaciju zaraze primjenjujući biljnu karantenu (prestanak trgovine zaraženim biljem, sjemenom i sadnim materijalom), koristeći pesticide te primjenjujući agrotehničke, mehaničke, fizičke i druge mjere. Implementacijom algoritama za rizik od obolijevanja, stvaraju se predviđanja koja daju informacije o potrebi i količini zaštite usjeva. Za stvaranje i procjenu ovakvih predikcijskih modela, ključna je Bayesova teorija [13].

2.1.2. Patogeni

Većina biljnih patogena ima izrazito velik reproduktivni potencijal stoga se njihova populacija drastično povećava u pogodnim vremenskim uvjetima. Patogeni mogu brzo razviti otpornost na pesticide i ostala kemijska sredstva kojima se pokušavaju suzbiti [14]. Također, brzo se prilagođavaju na okolišne i klimatske promjene. Kao takvi, predstavljaju veliku opasnost u infekciji biljnih organizama. Paul R. J. Birch et al. navode da biljke u cilju svoje zaštite, često hipersenzitivno reagiraju na napad patogena. Naime, izazivaju programiranu staničnu smrt koja je njihov urođen imunološki odgovor na napad [15].



Slika 2.2. Primjer patogena – mikroskopski prikaz pepelnice na listu krastavca [16]

2.1.3. Uspostavljanje dijagnoze

Prvi korak u učinkovitom uspostavljanju dijagnoze bolesne biljke je identifikacija biljnog organizma. Poznavajući vrstu biljke, često su poznati i patogeni koji ju mogu napasti. Također, važno je poznavati izgled i svojstva zdrave jedinice kako bi se mogla razlikovati od bolesne. U slučajevima kada biljku napadne više patogena, simptomi se mogu uvelike razlikovati od onih kada se bilježi prisutnost samo jednog patogena. Varijacije u simptomima mogu dovesti do netočnih dijagnoza stoga je važno laboratorijski ispitati uzorak [17]. Takve molekularne dijagnoze ostvaruju visoku točnost, međutim jako su spore i teško primjenjive na velikom uzorku [18]. U tu svrhu, implementiraju se modeli temelji na neuronskim mrežama koje kombiniraju svojsva lista, a treniranje i testiranje se provodi na bazi koja uključuje zdrave i bolesne primjere jedinki. Dodatno, u bazi se često nalaze i slike biljkaka koje rastu u laboratorijskim uvjetima te one tretirane pesticidima [19].

2.2. Primjena raspoznavanja bolesti lista biljke

Cilj izrade modela za prepoznavanje bolesti na listu biljke je, uz ranu detekciju, smanjiti trošak liječenja i spriječiti širenje bolesti na ostale usjeve [20]. Osvrt dijela znanstvenika na tadašnje metode za raspoznavanje bolesti biljaka davne 1936. godine, bio je:

“Potrebne su nam bolje dijagnostičke metode; nijedna od postojećih metoda ne smatra se dovoljno dobrom da bi postala 'standardizirana'. Razmišljajući o njima na ovakav način, nikada nećemo prestati ulagati napore u njihovo poboljšanje. One su korisne sve dok se ne razviju bolje metode.”

Riker i Riker [21]

Zbog rasta današnje populacije, pred preciznom poljoprivredom stavljeni su novi izazovi u svrhu veće i kvalitetnije proizvodnje hrane. Za ranije i preciznije uočavanje biljnih bolesti uz nemogućnost stalnog nadgledanja usjeva [20], potrebni su točniji modeli dubokog strojnog učenja, stvaranje sustava za udaljeno nadgledanje [20] te prilagodba upotrebe modela osobama bez stručnog računarskog znanja kroz korištenje interaktivne aplikacije [20, 22]. Takva aplikacija korisniku omogućuje unos slike bolesnih listova ili plodova biljke. Zatim se unose podaci o vrsti biljke i njenoj lokaciji kako bi proces prepoznavanja bio što točniji. U prepoznavanje bolesti su uključeni i vremenski podaci s naznakom na temperaturu i vlažnost. Također, mora postojati mogućnost unosa novih bolesti za što učinkovitiju zaštitu od epidemije [20].

Računalne dijagnoze uspješno prevladavaju teškoće u postavljanju dijagnoza za bolesti čija prisutnost predstavlja opasnost globalnih razmjera prilagođavanjem trenutnom zdravstvenom stanju pojedinih biljnih vrsta. Kao jedna od najvažnijih metoda računalne dijagnoze, duboko strojno učenje omogućuje brzu dostavu rezultata, manje troškove i porast učinkovitosti pri detekciji bolesti i lezija. Analizirajući podatke, algoritmi računalne dijagnoze također predviđaju i vjerojatnost pojave bolesti. Time predstavljaju dodatnu podršku stručnjacima pri donošenju odluka [23].

U eksperimentalnom dijelu rada korišteni su primjerci bolesnih i zdravih jedinki kukuruza. Siva pjegavost lista na kukuruzu, promatrana u ovome radu, uzrokovana je sporama na zaraženim ostacima kukuruzovine koje su nastale na površini tla. Zaraza započinje početkom lipnja kada se iz zaraženih biljnih ostataka oslobađaju spore koje vjetar raznosi na veće udaljenosti. Simptomi se pojavljuju u obliku velikih izduženih pjega čiji su rubovi tamniji i sive boje. Potom je promatrana obična hrđa koja na objema stranama plojke i rukavcu prouzrokuje smeđe jastučice što dovodi do žućenja listova. Vremenom, jastučici mogu puknuti te stvoriti crvene spore koje mogu zaraziti drugo lišće. Posljednja promatrana bolest je palež lista koja uzrokuje vodenaste pjege koje se brzo uvećavaju, a zatim i nekrotiziraju. Zaraženo tkivo se sužava i suši, a zaraza najčešće potječe iz zemljišta ili zaraženog sjemena.

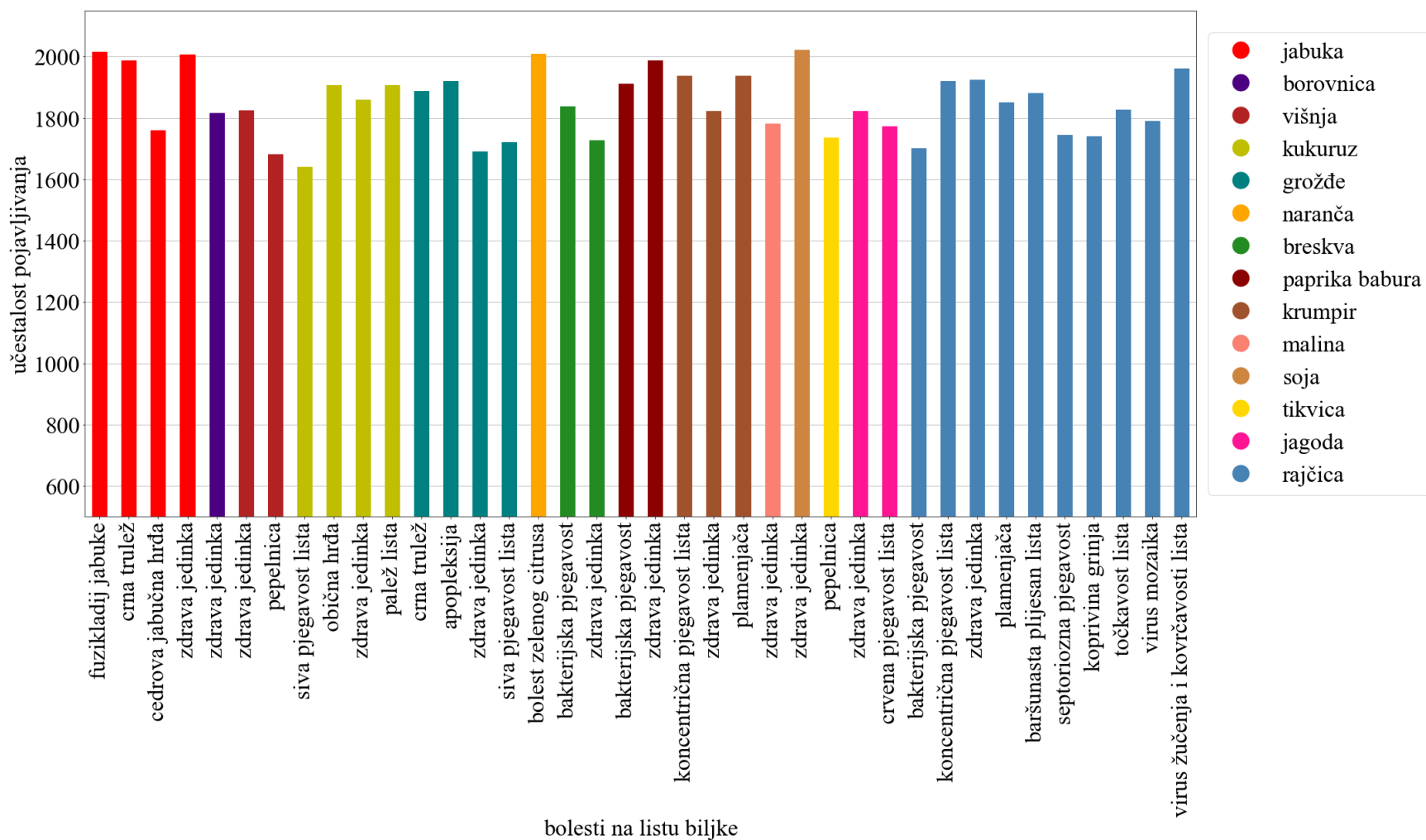
2.3. Baze slika biljaka za raspoznavanje bolesti

Najpoznatija baza podataka za raspoznavanje bolesti na listu biljke je *PlantVillage Dataset* koja se sastoji od slika u boji, crno-bijelih te segmentiranih slika. U ovome radu, korištena je prerađena inačica opisana u nastavku. Uz nju, dostupna je i baza podataka koja sadrži 18 222 primjeraka bolesnih jedinki kukuruza koja zbog postojanja samo jedne vrste bolesti, sive pjegavosti lista kukuruza nije primjenjiva za višeklasni problem u algoritmima dubokog strojnog učenja.

Korištena baza slika za treniranje i testiranje neuronskih mreža za raspoznavanje bolesti na listu biljke je *New Plant Diseases Dataset* [24]. Ova baza sadrži 87 867 različitih primjera zdravih i bolesnih listova biljaka koji su podijeljeni u 38 klasa, a na slici 2.3. prikazani su neki od njih. Za treniranje je korišteno 80% slika, dok je 20% slika korišteno za testiranje [24]. Podjela je provedena nasumično. Na slici 2.4. prikazana je približno ravnomjerna raspodjela trening primjera u klase, dok je s desne strane legenda koja označava pojedine vrste biljaka.



Slika 2.3. Primjeri slika iz baze



Slika 2.4. Distribucija trening primjera u 38 klasa s oznakom vrste biljke

2.4. Postojeća rješenja

Konstantinos P. Ferentios [25] je 2018. godine izradio modele za raspoznavanje bolesti na listu biljke temeljene na algoritmu konvolucijskih neuronskih mreža. Koristeći bazu slika s preko 80 000 primjeraka bolesnih i zdravih jedinki, ostvario je točnost od 99.53%. Uz primjerke listova biljaka u realnim uvjetima, dio baze činile su slike biljaka razvijene u laboratorijskim uvjetima. Sve slike su bile u boji jer su prijašnja istraživanja dokazala da crno-bijele slike nemaju bolje performanse u modelima dubokog strojnog učenja. Uspoređivao je performanse 5 osnovnih arhitektura: AlexNet (Krizhevsky et al., 2012.), AlexNetOWTBn (Krizhevsky, 2014.), GoogLeNet (Szegedy et al., 2015.), bOverfeat (Sermanet et al., 2013.) i VGG (Simonyan i Zisserman, 2014.) koja je imala najveću točnost. Rezultati su pokazali da modeli ostvaruju veću točnost kada se treniraju sa slikama biljaka koje rastu u realnim uvjetima, a testiraju slikama biljaka koje žive u laboratorijskim uvjetima i obratno [25].

Sharada P. Mohanty et al. [2] su 2016. godine koristeći bazu s preko 50 000 slika trenirali konvolucijsku neuronsku mrežu u cilju prepoznavanja 26 bolesti za 14 vrsta usjeva, ostvarivši točnost od 99.35%. Time su doprinjeli ostvarenju projekta za dijagnosticiranje bolesti biljke koristeći pametne uređaje na globalnoj razini. Baza podataka koju su koristili rezultat je projekta PlantVillage (Hughes i Salathé, 2015.). Koristili su 3 verzije baze slika: sa slikama u boji, s crno-bijelim slikama i slikama u kojima su izdvojili površinu lista, a pozadinu i ostale detalje zanemarili. Nakon treniranja, dokazali su da baza sa slikama u boji ima najveću, dok baza s crno-bijelim slikama ima najmanju točnost. Implementirali su dvije poznate arhitekture: AlexNet (Krizhevsky et al., 2012.) i GoogLeNet (Szegedy et al., 2015.) koja je dala bolje performanse. Iako naglašavaju da laboratorijska analiza daje točnije dijagnoze u odnosu na one koje su bazirane samo na vizualnim simptomima, smatraju da će dijagnoza pomoću pametnih uređaja i njihovih senzora uskoro dostići tu točnost [2].

Amanda Ramcharan et al. [26] su 2017. godine primijenili metodu prijenosnog učenja za treniranje duboke konvolucijske neuronske mreže u cilju prepoznavanja tri bolesti i dvije najčešće štete biljke cassava. Ova biljka predstavlja treći najveći izvor ugljikohidrata u supsaharskoj Africi. Koristeći bazu slika cassave snimljenih u Tanzaniji, najbolji model je ostvario točnost od 93%. Promatrajući različite genotipove i faze rasta cassave, prikupili su slike bolesnih listova cassave od početnih do krajnjih simptoma za pojedinu bolest. Za analizu performanse treniranja završnog

sloja Inception v3 modela konvolucijske neuronske mreže, koristili su tri različite arhitekture: inception softmax sloj, metodu potpornih vektora i metodu k-najbližih susjeda. Sličnu performansu su ostvarili inception softmax sloj i metoda potpornih vektora, dok se model k-najbližih susjeda (k=3) pokazao kao najlošiji. Razvijeni model su implementirali u android uređaje, a trenutno ih koriste znanstvenici u Tanzaniji za kontrolu rasprostranjenosti pojedinih bolesti [26].

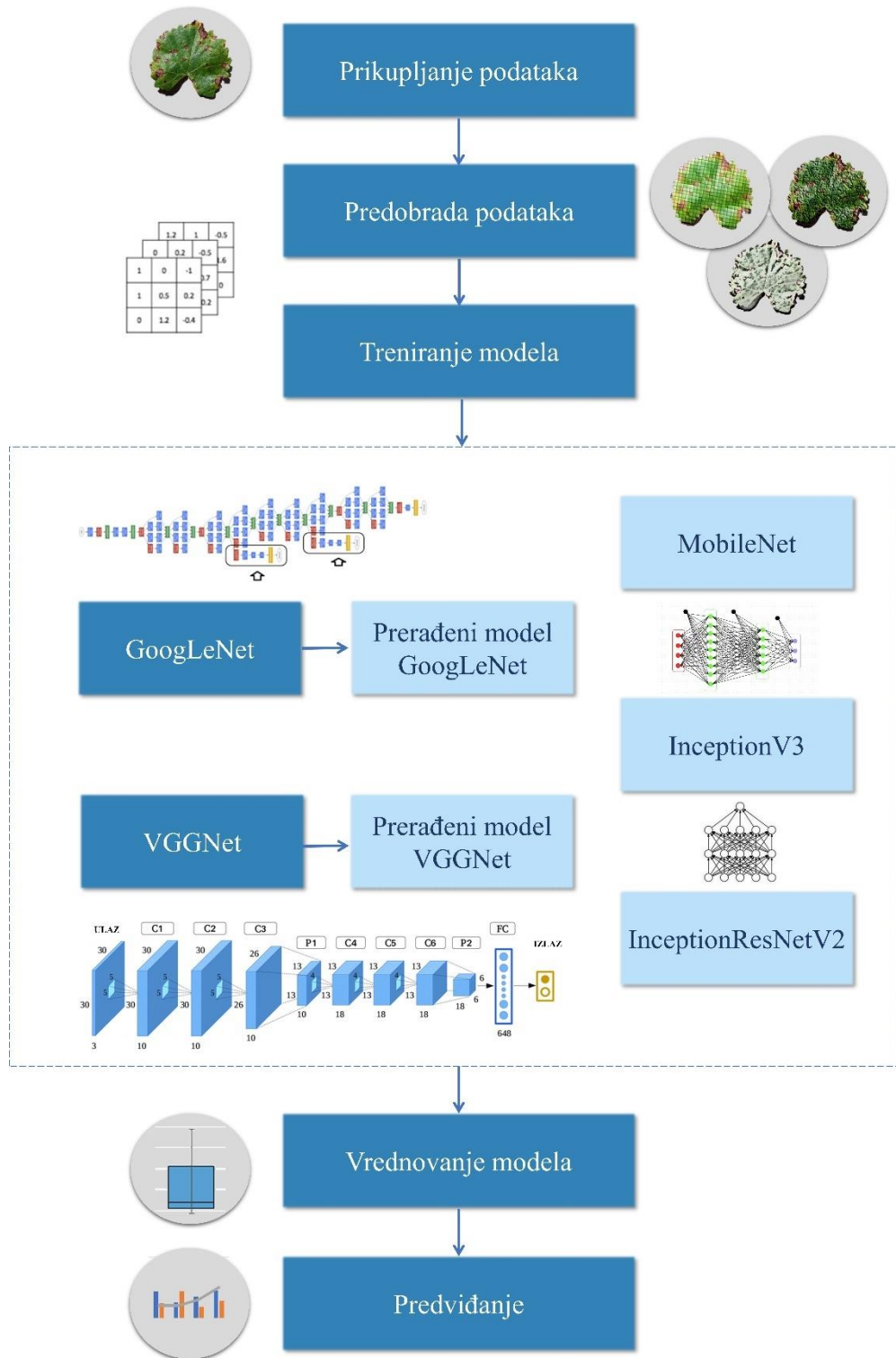
Koristeći prijenosno učenje i duboke konvolucijske neuronske mreže, Junde Chen et al. [27] su 2020. godine izradili model s ciljem ranog otkrivanja lezija na biljkama te biljnih bolesti uopće. Svoj model trenirali su i testirali na slikama zdravih i bolesnih listova biljaka dostupnima u PlantVillage (Hughes i Salathé, 2015.) bazi podataka koja je međunarodno korištena baza podataka za razvoj i primjenu algoritamskih rješenja upotrebom strojnog učenja. Također, model su testirali i na vlastito izrađenoj bazi podataka sa slikama bolesne riže i kukuruza. Kako bi izbjegli problem pretjeranog prilagođavanja podataka (engl. *overfitting*), slike su rotirali, povećavali im ili smanjivali razmjer kao i intenzitet osvjetljenja te ih predobradili. Korištenjem prethodno treniranog modela VGGNet, ostvarili su točnost od 91.83% za prepoznavanje bolesti dostupnih u PlantVillage bazi podataka i 92% točnosti za slike bolesne riže iz vlastito izrađene baze. Model VGGNet modificirali su zamjenom posljednjeg sloja dodatnim konvolucijskim slojem u kojem se kao aktivacijska funkcija, umjesto *ReLU*, koristila Swish funkcija. Konvolucijski sloj su zatim slijedila dva Inception modula, a potpuno povezani sloj zamijenjen je slojem sažimanja kako bi se smanjile dimenzije područja značajki. Nadalje, potpuno povezani softmax sloj postavljen je kao gornji sloj za ovu klasifikaciju. Uz daljnje razvijanje ovog modela, namjera im je implementirati ga u mobilne uređaje za prepoznavanje i praćenje biljnih bolesti [27].

3. POSTUPCI KLASIFIKACIJE KAO ALAT ZA RASPOZNAVANJE BOLESTI LISTA BILJKE

Razvoj modela dubokog strojnog učenja i računalnog vida potaknuo je eksponencijalni rast unaprjeđenja automatiziranih sustava za otkrivanje biljnih bolesti pomoću vidljivih simptoma na lišću [28]. Konvolucijska neuronska mreža, kao jedna od najpoznatijih metoda dubokog strojnog učenja, smatra se najperspektivnijim pristupom problemu raspoznavanja bolesti lista biljke [29] zbog sposobnosti pouzdanog izdvajanja vizualnih značajki [30]. Jedna od najvažnijih prednosti ove metode je mogućnost izdvajanja značajki bez upotrebe metoda segmentacije [29] koja je u ovome radu uz histograme za slike u boji korištena pri analizi ulaznih podataka. Korištenjem prijenosnog učenja omogućena je implementacija modela s visokom točnošću za specijaliziran skup podataka zbog preciznog podešavanja parametara prethodno treniranog modela na onom generalnom. Za slike bolesnih i zdravih listova biljaka koje rastu u kontroliranim uvjetima, dokazano je kako treniranje modela konvolucijskih neuronskih mreža uz prethodno trenirani model ostvaruje točnije rezultate od onog treniranoga po prvi put. Također, uklanjanje pozadine metodama segmentacije smanjuje učinkovitost treniranja mreže, time potvrđujući ovisnost pozadinskih značajki za identifikaciju bolesti [30]. U usporedbi s algoritmima klasičnog strojnog učenja, metode dubokog strojnog učenja ostvaruju zavidan napredak u točnosti i performansama modela. Primjerice, najboljih pet modela AlexNet arhitekture za ImageNet bazu podataka postiže pogrešku klasifikacije od 15.3%, dok isti podatak za modele klasičnog strojnog učenja iznosi 26.2% [31]. Proces izdvajanja značajki u metodama dubokog strojnog učenja ugrađen je u sam algoritam te je na taj način potpuno automatiziran. Iako algoritmi s ovakvim načinom izdvajanja značajki postižu zapaženo bolje rezultate, teško ih je interpretirati. U tu svrhu, koriste se metode vizualizacije [31]. Izdvajanje značajki dubokim strojnim učenjem proučavao je manji broj studija, no sam postupak i dalje ostaje nepoznat (engl. *black box*). Znanje o ovom procesu značajno bi doprinijelo području znanosti pri razumijevanju dijagnostičkih postupaka povećavajući učinkovitost uporabe dubokog strojnog učenja pri detekciji biljnih bolesti [32].

U ovome poglavlju, uz pripadajuće materijale i korištene metode, opisane su potrebne tehnike predobrade podataka uz njihovo prikupljanje. Kao klasifikatori, predstavljene su neuronske mreže s naglaskom na konvolucijske neuronske mreže. Prikazan je opis korištenih mreža i njihovih

arhitektura. Nakon objašnjenih postupaka analize, opisana je usporedba performansi. Na slici 3.1. prikazani su korišteni materijali i metode u ovome radu uz tijek rada.



Slika 3.1. Materijali i metode za raspoznavanje bolesti na listu biljke

3.1. Prikupljanje podataka i predobrada

Tehnička ograničenja prilikom treninga modela vezana uz automatsku klasifikaciju bolesti biljaka uglavnom su gotovo neprimjetna u prikazu performansi dok uporaba ograničenog skupa podataka donosi mnogo neželjenih posljedica i netočnih rezultata predviđanja. Kako bi se izbjegao ovaj problem, potrebno je osigurati konzistentnost baze podataka koja u ovom radu sadrži primjerke bolesnih listova biljke ne uzimajući u obzir moguću zahvaćenost cvijeta, stabljike ili plodova. Također, potrebno je osigurati primjerke slika biljaka koje se nalaze u raznim životnim uvjetima, poput onih u laboratorijima, kontroliranim ili realnim uvjetima [33].

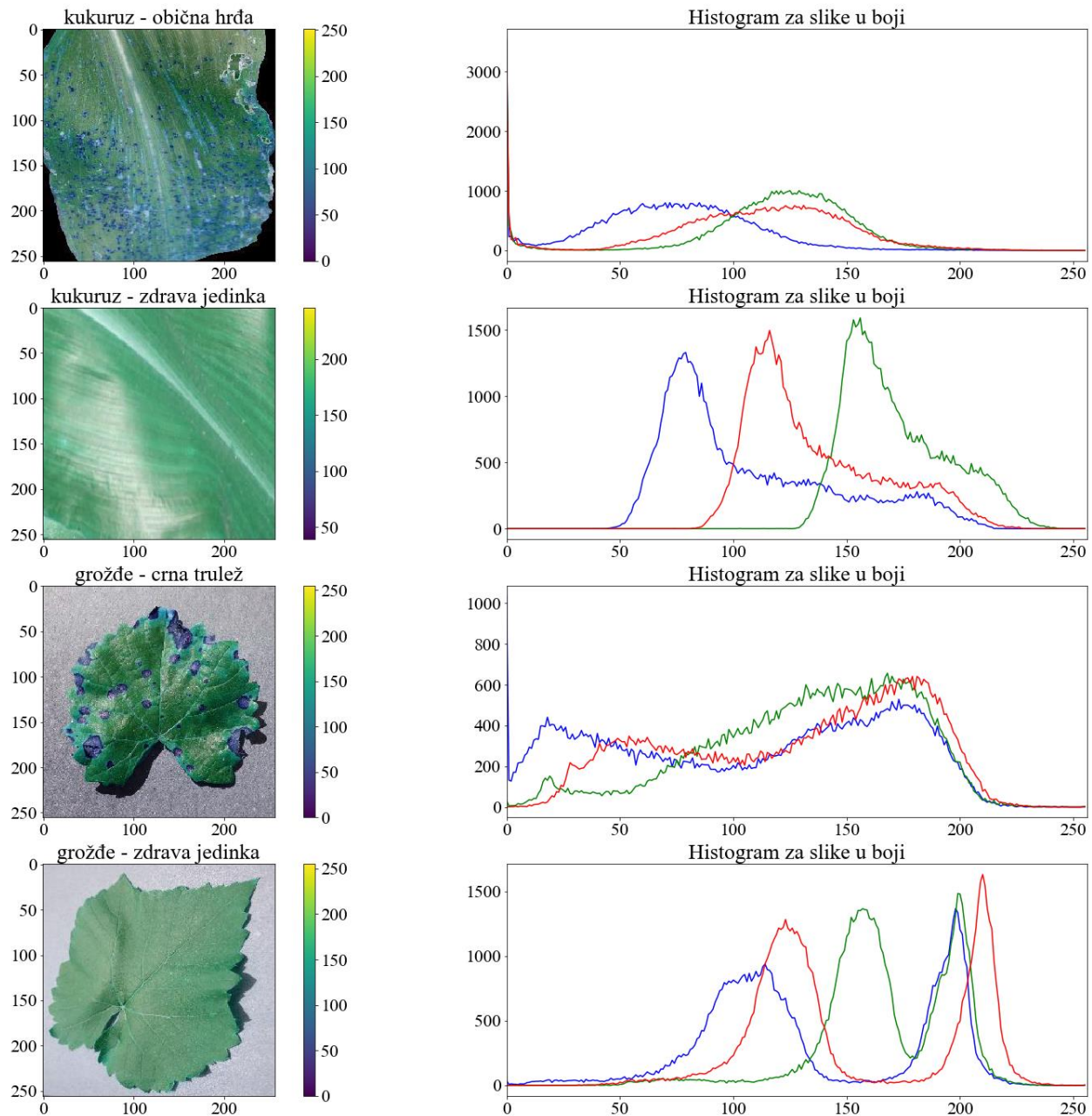
Sve slike u korištenoj bazi nalaze se u RGB formatu s vrijednošću piksela 256x256 te s vrijednošću od 24 bita po pikselu gdje je svaki piksel sastavljen od 8-bitnog crvenog, zelenog i plavog kanala. Vrijednost jedinice bit po pikselu označava ukupnu razinu boje sadržane u svakom pikselu u kojoj primjerice 24 bita po pikselu predstavlja najveću vrijednost čiji je skup boja vidljiv ljudskom oku [34].

Barbedo [33] navodi kako u teoriji, uklanjanje pozadine kao postupak predobrade slika ne bi trebao negativno utjecati na rezultate dobivene izvedbom klasifikacije. Međutim, prilikom takvog postupka primijetio je kako konvolucijska neuronska mreža pri klasifikaciji bolesti biljaka može koristiti i pozadinske karakteristike slike umjesto samo simptomatskih. Napominje kako do istog rezultata može doći pri različitim fazama razvoja bolesti uz pojavu drugačijih simptoma od onih prisutnih u bazi [33].

Za potrebe segmentacije slika i ubrzanje računskih operacija, a kako bi se provelo istraživanje i usporedile performanse modela koristeći različite metode strojnog učenja među kojima su i konvolucijske neuronske mreže, Sharma et al. su smanjili veličinu piksela te uklonili prisutnu buku prebacivši slike iz RGB formata u format HSV čiji spektar boja sadržava manje šuma [35]. Prilikom stvaranja baze slika, poznato je kako neuravnotežena baza ili ona s manjim brojem slika najčešće dovodi do pretjeranog prilagođavanja podataka onoj klasi koja sadrži najviše podataka. Pri rješavanju ovog problema, primjenjuju se postupci transformacije poput rotacije slika, njihovo uvećavanje te promjena svjetline i razine izoštrenosti. Cilj je nasumičnim mijenjanjem ovih karakteristika, doprinijeti povećanju baze brojem slika koje je moguće pronaći u stvarnosti pri različitim životnim uvjetima [36]. Takvi primjerci dio su *New Plant Diseases Dataset* [24], korištene u ovome radu.

3.1.1. Histogram za slike u boji

Za digitalne slike, histogram za slike u boji predstavlja grafički prikaz distribucije intenziteta piksela [37, 38] gdje su za 3 kanala R, G i B izrađena 3 histograma [38]. Histogrami su relativno nepromjenjivi prilikom translacije ili rotacije slike, a manje varijacije su moguće tek prilikom različitog kuta gledanja [37]. Os apscisa poprima vrijednosti piksela koji su za slike u boji u intervalu [0, 255], dok os ordinata prikazuje broj pojavljivanja određenog intenziteta vrijednosti na slici [37].

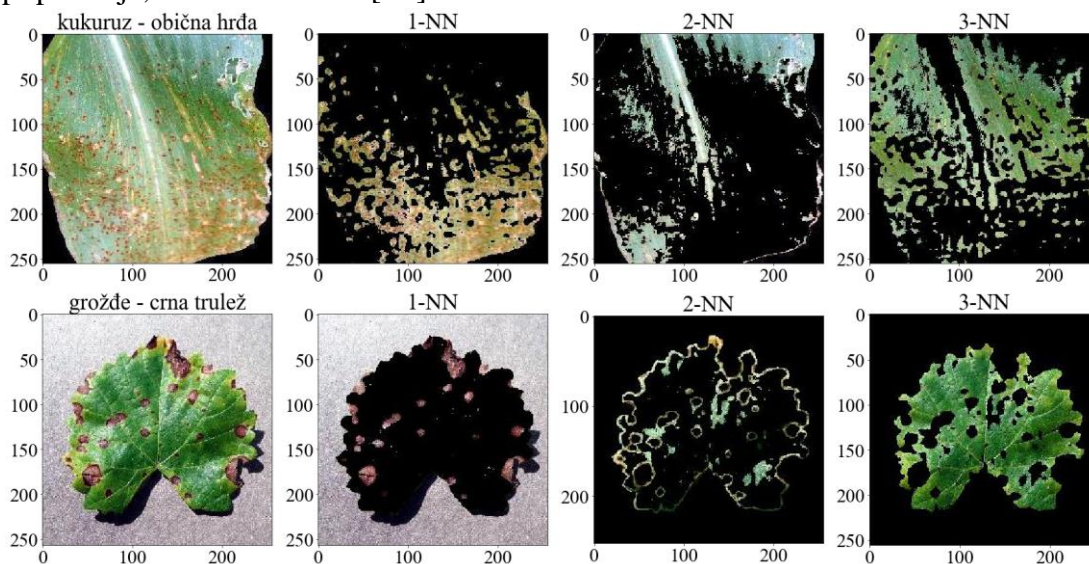


Slika 2.5. Usporedba histograma u boji za zdrav i bolestan list kukuruza i grožđa

Usporedba histograma dvije slike omogućila je klasificiranje i prepoznavanje objekata [37], a često korištena metoda za prepoznavanje bolesti na listu biljke je uklanjanje piksela s vrijednostima zelene boje [39]. Na slici 2.5. prikazana je usporedba histograma u boji za zdrave i bolesne uzorke lista iz baze, za kukuruz i grožđe. Kod zdravih jedinki, os ordinata poprima veće vrijednosti što znači da je cijela površina lista približno iste, zelene boje. Kod bolesnih jedinki te vrijednosti su manje što možemo protumačiti kao prisutnost drugačijeg intenziteta više boja na površini lista, ili prisutnost bolesti. Po kanalima, vrijednosti na osi ordinata su približno jednake za primjer bolesne jedinice.

3.1.2. Segmentacija slika algoritmom k-najbližih susjeda (k-NN)

Osim što je brz, robustan i lako prilagodljiv, algoritam k-najbližih susjeda jedan je od najčešće korištenih algoritama za segmentaciju slika te višeklasnu klasifikaciju [40]. Radi se o metodi nenadziranog strojnog učenja koja može klasificirati podatke bez prethodnog treninga pronalaskom k najbližih susjeda koji su određeni središnjom točkom (engl. *centroid*) i njihovih klasa. Metoda računa udaljenost između novog i svakog trening podatka s ciljem pronalaska najbližeg susjeda (ili njih k, gdje je k element skupa prirodnih brojeva) promatranom podatku. Izračuni su najčešće temeljni na metodi Euklidske udaljenosti. Ova metoda korištena je za segmentaciju slika zbog preciznog raspoznavanja i odvajanja pozadine slike te bolesnog i zdravog dijela lista biljke [41]. Prilikom segmentacije, nedostatak algoritma k-najbližih susjeda je egzaktno određivanje središnje točke i opcionalnog broja susjeda zbog složenosti značajki slike bolesnog lista poput boje, oblika i teksture [42].



Slika 3.6. Segmentacija bolesnih listova kukuruza i grožđa algoritmom k-najbližih susjeda

Rezultati prerađenog algoritma k-NN [43] za 3 najbliža susjeda prikazani su na slici 2.6. Udio lista kukuruza koji je zahvaćen bolešću iznosi 15.2459%, dok je za list grožđa taj podatak jednak 20.5274%.

3.2. Klasifikatori

Metode klasičnog strojnog učenja imaju svojevrсно ograničenje pri izdvajanju različitih značajki iz trening skupa podataka. Taj nedostatak uspješno prevladavaju klasifikatori dubokog strojnog učenja koji imaju slojevitu strukturu poput umjetnog neuronskog sustava koji je nalik biološkoj neuronskoj mreži ljudskog mozga [44].

Jedan od problema računalnog vida je klasifikacija slika, točnije stvaranje algoritama i modela koji kao ulazni podatak zahtijevaju sliku za koju na izlazu vraćaju skup kategorija koji ju najbolje opisuju ili njezinu pripadnost pojedinom razredu. Izražava se funkcijom

$$f: \mathbb{R}^n \rightarrow \{1, \dots, k\} \quad (3 - 1)$$

u kojoj je k prirodni broj. Ako vrijedi sljedeća relacija, model ulazu x dodjeljuje izlaznu kategoriju y [45].

$$y = f(x) \quad (3 - 2)$$

Navedeni primjer pripada nadziranom strojnom učenju koje za trening skup unaprijed zna očekivane vrijednosti izlaza koje su diskretne, a cilj joj je pronaći funkciju koja je najučinkovitija i najtočnija. Kako bi se provjerila točnost te funkcije, klasifikator isti postupak obavlja na testnom skupu podataka koji su mu dosad bili nepoznati [46].

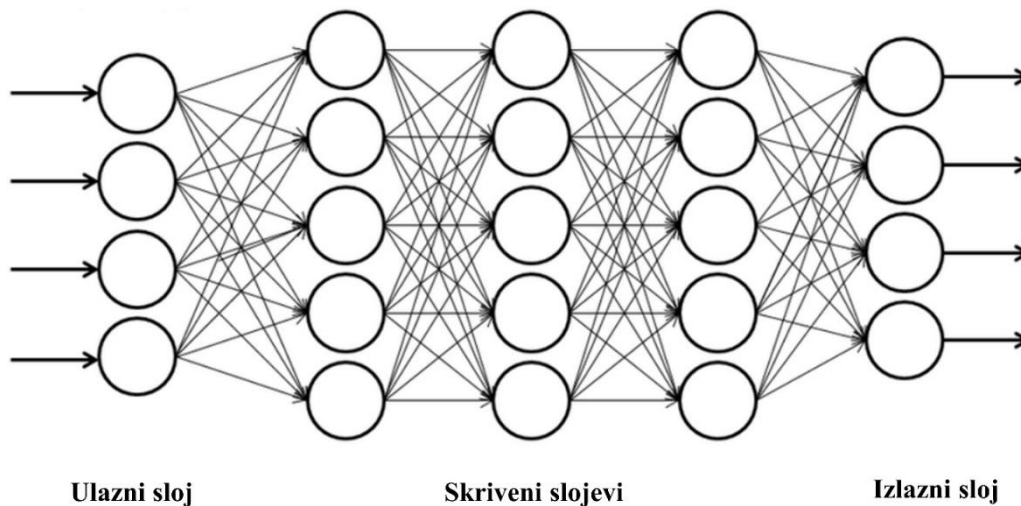
3.2.1. Neuronska mreža

Skup algoritama dizajniranih s ciljem prepoznavanja obrazaca, ili neuronske mreže, interpretiraju podatke pomoću vrste strojne percepcije označavajući ili grupirajući podatkovni unos. Pri tome, obrasci se nalaze u numeričkom obliku, a sadržani su u vektorima u kojima moraju biti prevedeni svi signali iz realnog svijeta poput slika, zvuka, videa ili teksta. Osim što grupiraju nepoznate ulazne podatke prema sličnostima, mogu obavljati klasifikaciju ako postoji označeni skup podataka za trening. Nadalje, neuronske mreže mogu se nalaziti u ulozi posrednika prilikom kojeg stvaraju novi skup podataka na temelju izdvojenih značajki kojeg potom predaju drugom algoritmu strojnog učenja [47, 48].

Primjenjuju se prilikom procesa prepoznavanja govora i slike, medicinskim dijagnozama, stvaranju novih informacija prilikom ispitivanja brojnih baza podataka, detektiranju anomalija te su neizostavna pomoć stručnjacima u donošenju odluka za različite grane znanosti [48].

U pokušaju oponašanja biološke organizacije i hijerarhije neuronskih mreža, Hinton et al. su 2018. godine ostvarili poboljšanja u modelu mreža kapsula, a koji je vrsta umjetne neuronske mreže strojnog učenja. Ideja je u neuronsku mrežu dodati strukture nazvane kapsule koje u sebi sadrže izlazni vektor mreže čije vrijednosti ponovno iskorištavaju stvarajući nove, sadržajnije kapsule smanjujući početne smetnje. Kapsula je tako skup neurona koji otkrivaju različita svojstva objekata poput njegova položaja ili veličine. Donose prednosti u rješavanju Piccasovog problema, uspješno prepoznavajući slike čiji dijelovi nisu u ispravnom prostornom odnosu usmjeravajući sve informacije iz kapsula niže razine u najbliže kapsule na višim razinama mreže [49].

Sastavljena od najmanje četiri razine slojeva koje sadrže čvorove, neuronska mreža klasificira se kao metoda dubokog strojnog učenja, za razliku od jednostavnih perceptona koji se sastoje od ulaznog i izlaznog sloja te najviše jednog skrivenog sloja između njih. U dubokim neuronskim mrežama, svaki sloj čvorova obavlja operacije na različitim skupinama značajki temeljem izračunatih rezultata prethodnog sloja. Povećanjem broja slojeva i hijerarhije značajki, povećava se apstrakcija mreže. Ovo svojstvo stvara sposobnost obrađivanja brojnog i višedimenzionalnog skupa podataka s brojnim parametrima koji postaju dijelovi nelinearnih funkcija [47]. Primjer duboke neuronske mreže koja je uz ulazni i izlazni sloj sastavljena još od tri skrivena sloja prikazan je na slici 3.2.



Slika 3.2. Model duboke neuronske mreže [50]

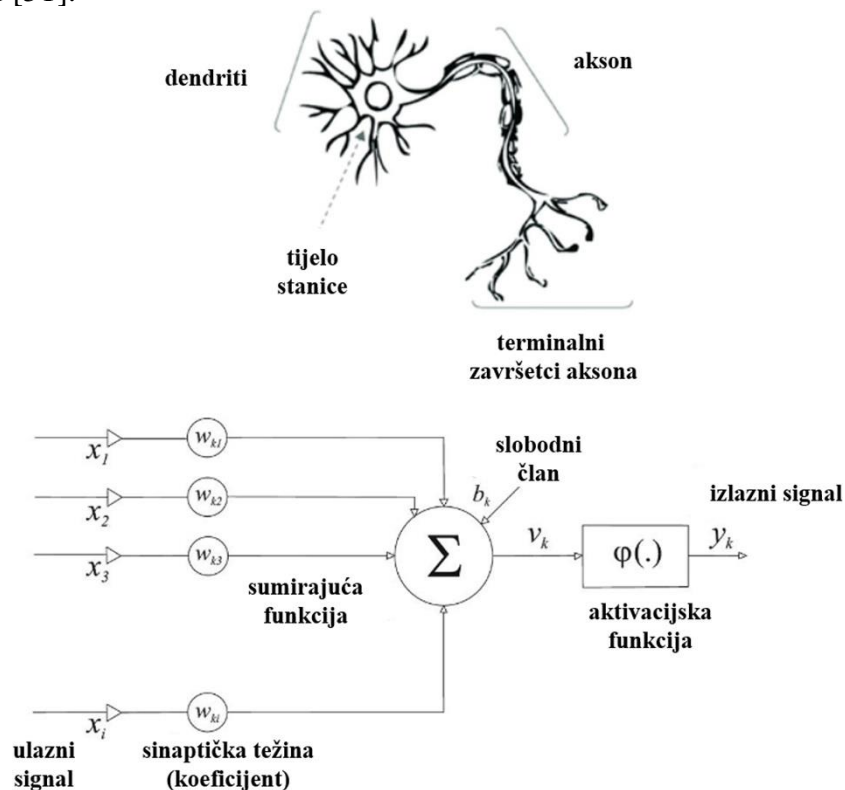
Zadaća svih umjetnih neurona je proces obrade i prijenosa informacije prilikom kojeg oponašaju biološke neurone, sastavne dijelove živčanog sustava. Tijekom ovog procesa, ulazni signali x_1, x_2, \dots, x_i povezani su sinaptičkom težinom w_{ki} , točnije vremenskom funkcijom i slobodnim koeficijentom b_k (engl. *bias*) koja predstavlja graničnu vrijednost (engl. *threshold*) prilikom iscertavanja karti značajki. Nakon sumiranja vrijednosti ulaznih signala i sinaptičkih težina, dobije se ulaz mreže u_k :

$$u_k = \sum_{j=1}^i w_{kj} x_j \quad (3 - 3)$$

na koji se pribroji slobodni koeficijent čineći aktivacijski potencijal neurona v_k . Svaki pojedinačni sloj mreže prima ulazne signale smještene u prethodnom sloju. Kako bi se dobio izlaz mreže, potrebno je djelovanje aktivacijske funkcije $\varphi(\cdot)$ na aktivacijski potencijal neurona koji je opisan sljedećom jednačbom [51]:

$$y_k = \varphi(u_k + b_k) \quad (3 - 4)$$

Aktivacijska funkcija daje neuronu nelinearno svojstvo koje omogućava modeliranje složenih nelinearnih skupova podataka [52]. Proizvedeni izlazni signal dalje se prenosi u neurone u sljedećem sloju [51].



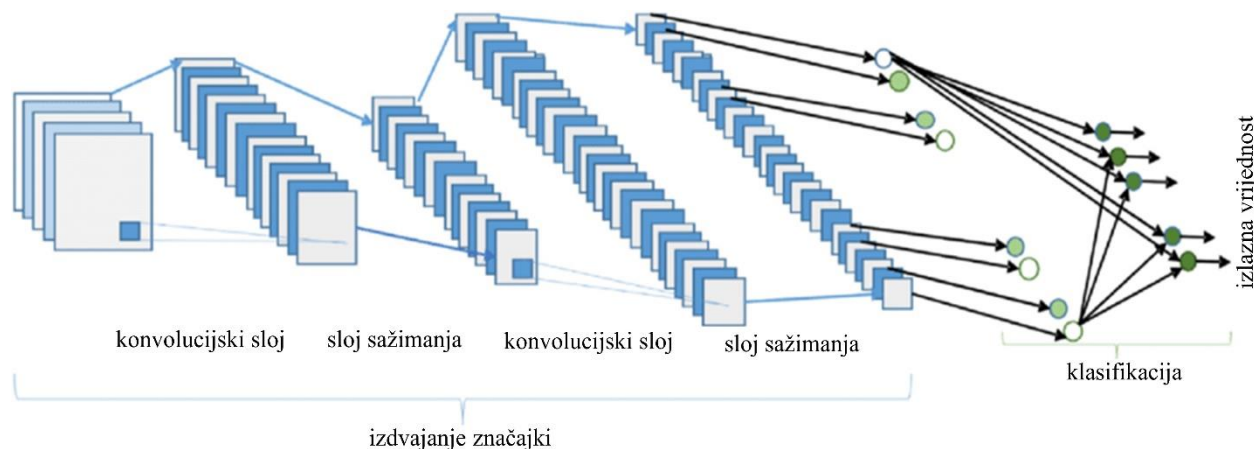
Slika 3.3.

biološkog i umjetnog neurona [51, 53]

Model

3.2.2. Konvolucijska neuronska mreža – CNN (engl. *Convolutional neural network*)

Imitirajući funkcije nalik onima u ljudskom vidnom sustavu, CNN struktura izdvaja značajke obavljajući operaciju konvolucije na ulaznom podatku, potom izdvaja maksimalne ili prosječne vrijednosti tih značajki, a sve ih spaja potpuno povezani sloj dajući rezultat klasifikacije [54]. Dodatno, između konvolucijskih slojeva djeluje aktivacijska funkcija koja predstavlja svojevrsni filter prilikom izlaza iz trenutnog neurona, a pri ulazu u sljedeći. Primjer takve arhitekture prikazan je na slici 3.4. Za ubrzanje konvergencije te kontroliranje treniranja neuronske mreže često se koriste takozvani „*batch normalization*“ ili „*dropout*“ slojevi [55].



Slika 3.4. Primjer CNN arhitekture [56]

Prilikom izračunavanja vrijednosti i ocjenjivanja performanse modela, funkcija izračuna gubitka (engl. *loss function*) prilikom treniranja kao parametre uzima izlaz modela i uspoređuje ga s ciljanom vrijednošću te daje rezultat. Djeluje u dva smjera: prolaz naprijed prilikom kojeg se izračunava vrijednost gubitka ovisno o ulaznoj i ciljanoj vrijednosti te prolaz natrag kojim se izračunava gradijent *loss* funkcije povezane s određenim kriterijem [57].

Prolaskom natrag, koristi se algoritam propagacije pogreške unatrag (engl. *backpropagation algorithm*) koji izračunavajući gradijent na svakom sloju mreže predstavlja učinkovitiji pristup od onog tradicionalnog [57]. Nadalje, algoritam se prilagođava različitim težinama modela. Prvotno se određuje gradijent posljednjeg sloja, potom se pri određivanju gradijenta prethodnog sloja koriste djelomični proračuni gradijenta trenutnog sloja, a posljednje se izračunava gradijent prvog sloja konvolucijske neuronske mreže [58].

3.2.2.1. Konvolucija

Prema [59], konvolucija se definira kao matematička funkcija nastala integriranjem umnoška dviju funkcija po intervalu njihove definicije gdje su te funkcije ravnopravne tako da svaka infinitezimalna promjena jedne funkcije utječe na drugu funkciju u cijelome intervalu definicije. Primjenom konvolucije dolazi do smanjenja smetnji kod ulaznog signala koji postaje otporniji na buku [60]. Konvolucijska neuronska mreža pri tome koristi diskretnu konvoluciju:

$$(u * v)(n) = \int_{m=-\infty}^{\infty} u(n - m)v(m)dt \quad (3 - 5)$$

gdje su n i m konačni broj koraka uz signale u i v , a operacija konvolucije prikazana je znakom $*$ [57].

Rezultat ove operacije pohranjuje se u sklopovsku jedinicu registra akumulatora [61]. U nastavku je prikazan opis algoritma matematičke operacije konvolucije u pseudokodu, a postupak je prikazan i na slici 3.5. na primjeru ulazne slike veličine 7×7 i filtera (engl. *kernel*) veličine 3×3 .

Algoritam 1 Matematička operacija konvolucija [57]

for each red matrice slike **in** ulazna slika:

for each piksel **in** red matrice slike:

set akumulator na nula

for each red konvolucijske matrice **in** konvolucijska matrica:

for each element **in** red konvolucijske matrice:

if pozicija elementa odgovara poziciji piksela **then**

multiply vrijednost elementa za odgovarajuću vrijednost piksela

add rezultat u akumulator

end if

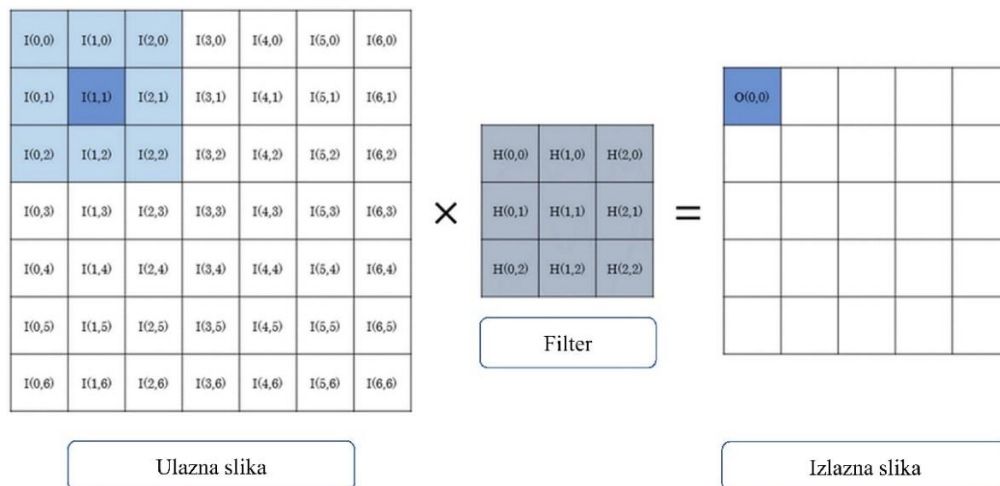
end for

end for

end for

end for

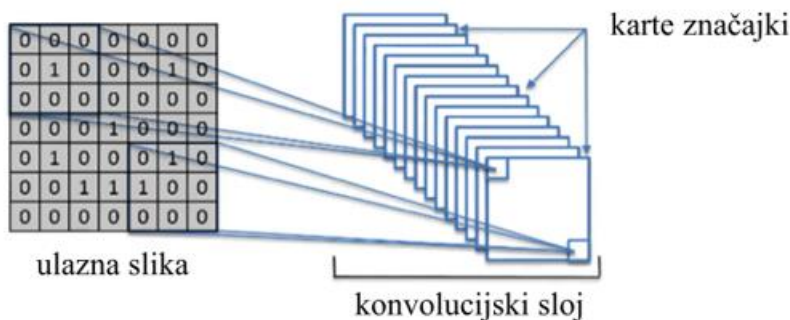
set piksele izlazne slike na akumulator



Slika 3.5. Konvolucija na primjeru ulazne slike veličine 7x7 i filtera veličine 3x3 [62]

3.2.2.2. Konvolucijski sloj

Iako CNN modeli ostvaruju veoma uspješnu točnost prilikom prepoznavanja objekata, optimalna arhitektura kao i najučinkovitija implementacija konvolucijskih slojeva i dalje se istražuje [63]. Prvi sloj ujedno je i najvažniji u stvaranju konvolucijskih neuronskih mreža koji koristeći filtere obavlja operaciju konvolucije i stvara pripadajuće karte značajki. Broj filtera određuje broj detektora značajki, ali i povećava mogućnost pretjeranog prilagođavanja podataka [64].



Slika 3.6. Uloga konvolucijskog sloja u modelu konvolucijskih neuronskih mreža [65]

Određuju ga parametri poput dubine izlaznog volumena koja kontrolira broj neurona u sloju koji se povezuju na isto područje ulaznog volumena. Svi neuroni aktivirat će se za različite značajke ulazne slike. Nadalje, korak pomaka filtera određuje prostornu dubinu stupaca zahvaćenu filterom. Primjerice, ako korak pomaka filtera ima vrijednost 1, tada filter prelazi jedan po jedan piksel slike. Rezultirajući volumen imat će manje prostorne dimenzije ako je preklopljenost receptivnih

polja manja. Posljednji parametar određuje širenje ulaznog volumena poljima koja sadrže nule (engl. *zero-padding*), a u cilju očuvanja prostornih dimenzija i dizajniranja dublje mreže [57]. Sprječava smanjenje karte značajki tijekom konvolucije jer se sada središnji piksel filtera pozicionira u rubni piksel ulazne slike [64].

Izlaz konvolucijskog sloja može se izraziti sljedećom jednačinom:

$$M_j^p = f \left(\sum_{i \in M_j} M_i^{p-1} * k_{ij}^p + N_j^p \right) \quad (3 - 6)$$

u kojoj M_j predstavlja skup ulaznih karti značajki, p označava redni p -ti broj sloja dok je k_{ij} konvolucijska matrica, a N_j slobodni koeficijent. Vrijednosti parametara poput težine konvolucijske matrice te slobodnog koeficijenta rezultat su treniranja mreže metodama nenadziranog strojnog učenja [66].

Konvolucijski sloj također omogućava invarijantnost prostorne pozicije objekta, lokalnu povezanost koja uči korelacije između susjednih piksela te mehanizam dijeljenja težine na istoj karti značajki smanjujući ukupan broj parametara [57].

Kao i ostale neuronske mreže, CNN uključuje uporabu nelinearne aktivacijske funkcije nakon izračuna provedenih u konvolucijskom sloju. Obično se ona definira unutar samog konvolucijskog sloja, no moguće ju je implementirati i u zaseban sloj radi omogućavanja veće fleksibilnosti mrežne arhitekture modela [64].

3.2.2.3. Aktivacijska funkcija – *ReLU* (engl. *Rectified Linear Unit*)

Svaka matematička operacija koja se izvodi tijekom treniranja mreže uključuje linearnu transformaciju koju slijedi aktivacijska funkcija uvodeći nelinearnost u mrežu omogućavajući učenje složenih skupova podataka, izračunavanja te predviđanja. Uz navedeno, aktivacijska funkcija zauzima integralnu ulogu u procesu treniranja i ocjenjivanja performansi mreže [67, 68]. Svojevrsni je matematički filter između trenutnog neurona i izlaza koji prelazi u idući sloj [69]. Jedna od funkcija koja predstavlja standard za svaku primjenu je *ReLU* aktivacijska funkcija definirana:

$$f(x) = \max(0, x) \quad (3 - 7)$$

gdje je povratna vrijednost 0 ako je ulazni podatak negativan, a svaka pozitivna vrijednost koja je predana, ujedno je i vraćena. Domena je ograničena na intervalu $[\approx -0.31, \infty)$ [67, 68].

Računalno je učinkovita, omogućavajući brzo konvergiranje mreže i provođenje algoritma propagacije pogreške unatrag osim kada gradijent funkcije ima vrijednost nula [69]. Ovu funkciju moguće je zamijeniti skalarnom koja kao ulazni i izlazni podatak uzima brojčanu vrijednost ne mijenjajući arhitekturu mreže [68].

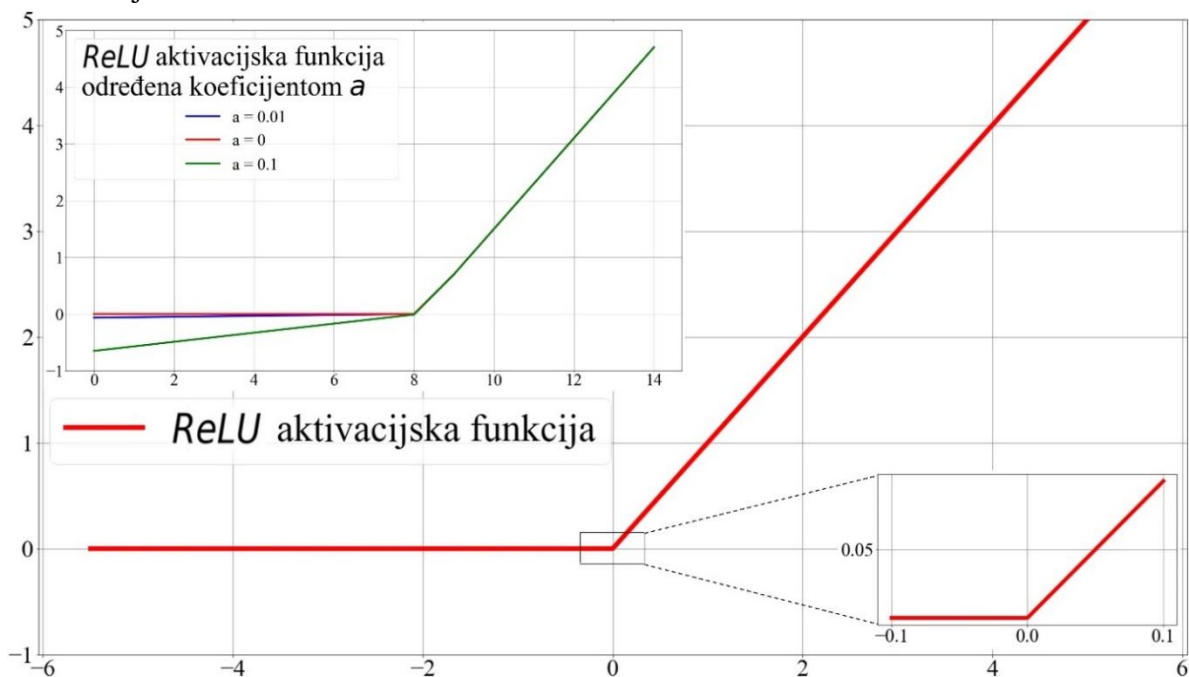
Najkorištenije inačice ove funkcije su propustljiva te parametarska *ReLU* funkcija. Definirane su izrazom:

$$f(x) = \max(ax, x) \quad (3 - 8)$$

u kojem je s a označen koeficijent propuštanja koji kod propustljive *ReLU* funkcije rezultira malim nagibom u negativnom području omogućavajući provođenje *backpropagation* algoritma. Međutim, za negativne ulazne vrijednosti funkcija ne ostvaruje uvijek dosljedna predviđanja [69].

Za razliku od propustljive, kod parametarske *ReLU* funkcije koeficijent propuštanja se određuje prilikom treniranja mreže, a sadrži najprikladniju vrijednost ovisno o arhitekturi mreže i dostupnoj bazi podataka na kojoj se model trenira [69].

Na slici 3.7. grafički su prikazane prethodno navedene funkcije određene koeficijentima propuštanja s vrijednostima $a = 0.01$ za propustljivu *ReLU* funkciju, $a = 0$ za *ReLU* funkciju te za proizvoljno odabranu vrijednost koeficijenta $a = 0.1$ kao primjer koji određuje parametarsku *ReLU* funkciju.



Slika 3.7. Grafički prikaz *ReLU* aktivacijske funkcije

3.2.2.4. Swish i Mish aktivacijske funkcije

Prilikom treniranja neuronskih mreža, dokazano je kako Swish aktivacijska funkcija nadmašuje performanse *ReLU* ili ih, iznimno, barem dosljedno prati. Ramachandran et al. navode kako Swish povećava preciznost top-1 klasifikacije za 0.9%, a istraživanje je provedeno na ImageNet bazi podataka.

Zaglađena je i nemonotona te ograničena na istom intervalu poput *ReLU* funkcije. Izraz koji ju definira glasi:

$$f(x) = x * \text{sigmoid}(\beta x) \quad (3 - 9)$$

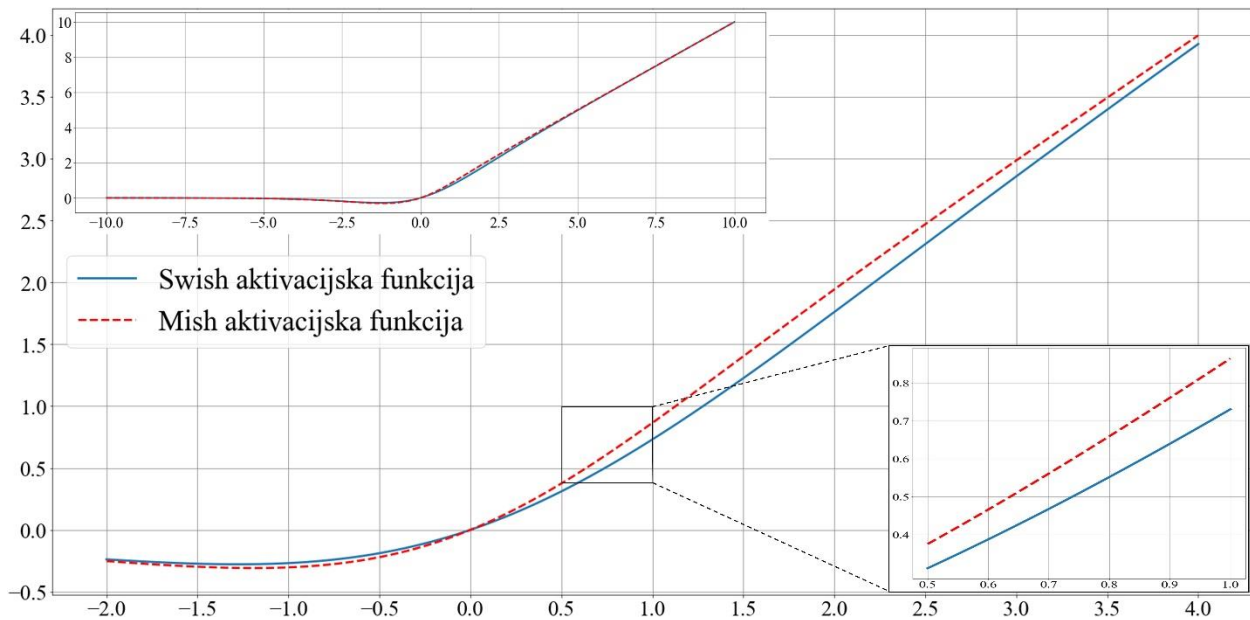
gdje je β konstanta ili parametar određen treniranjem mreže za sigmoidnu funkciju. Kada koeficijent β teži u beskonačnost, Swish poprima oblik *ReLU* aktivacijske funkcije. Točnije, moguće ju je promatrati kao funkciju koja interpolira između linearne i *ReLU* funkcije. Ako je β koeficijent čija se vrijednost određuje treniranjem mreže, tada model ujedno i određuje razinu interpolacije. Najvažnija razlika između Swish i *ReLU* funkcije nemonotonični je „skok“ u Swish funkciji kada x poprima negativne vrijednosti, a nalazi se na području predaktivacijskog procesa [68].

Treniranjem dubokih mreža na zahtjevnim skupovima podataka uočeno je kako modeli koji koriste Mish aktivacijsku funkciju ostvaruju točnije rezultate od onih koji implementiraju *ReLU*, Swish i ostale standardne aktivacijske funkcije. Po definciji, određuje ju sličnost sa Swish funkcijom, a zajedno s većim performansama i boljim učinkom te jednostavnom izvedbom najčešći je odabir pri implementaciji neuronskih mreža. Misra [67] navodi kako Mish aktivacijska funkcija povećava preciznost top-1 klasifikacije za 0.494% uspoređujući ju sa Swish funkcijom te za 1.671% uspoređujući ju s *ReLU*, a model je treniran na *CIFAR 100* bazi podataka. Zbog navedenog, preporučuje se određivanje niže brzine učenja (engl. *learning rate*) pri treniranju modela.

Po svojstavima poput zaglađenosti i nemonotoničnosti te jednostavnoj implementaciji slična je Swish funkciji, a definirana je izrazom:

$$f(x) = x * \text{tanh}(\text{softplus}(x)) = x * \text{tanh}(\ln(1 + e^x)) \quad (3 - 10)$$

pri čemu je ograničena na istome intervalu kao i prethodno navedene funkcije, *ReLU* te Swish [69]. Usporedba Mish i Swish aktivacijske funkcije prikazana je na slici 3.8.



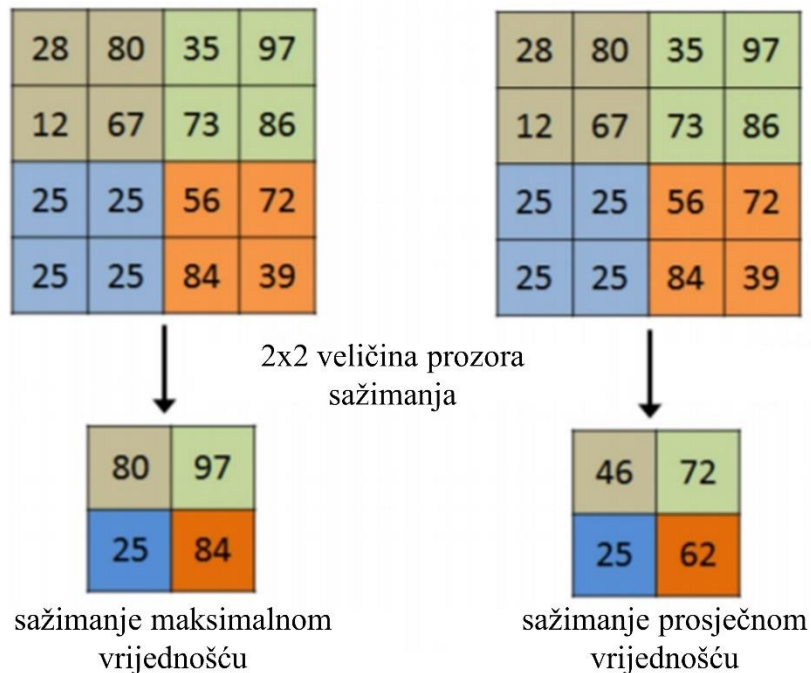
Slika 3.8. Grafička usporedba Swish i Mish aktivacijskih funkcija

Po uzoru na Swish, ova funkcija pripada skupini *self-gating* funkcija provodeći *gating* mehanizam u kojem se ista vrijednost upotrebljava za sami *gate*. Prednost ovog mehanizma korištenje je jednogulaznog skalara dok se kod uobičajenog mehanizma zahtijeva korištenje više njih [68]. Ovo svojstvo također omogućava provođenje *backpropagation* algoritma na gradijentima [70].

3.2.2.5. Sloj sažimanja

Upotrebom sloja sažimanja u CNN modelima, provodi se pojednostavljivanje ili smanjivanje prostornih dimenzija informacije dobivene iz karte značajki. Najčešće se koristi proces sažimanja maksimalnom vrijednošću zbog brzine i poboljšane konvergencije, a značajan je i proces sažimanja s prosječnom vrijednošću te *L2-norm* vrsta sloja sažimanja ovisno o primjeni. Obično se pomoću filtera veličine 2x2 uz korak pomaka filtera iste dimenzije prelazi preko ulaznog volumena, a rezultat je maksimalan ili prosječni broj svake podregije [57]. Navedeni postupak prikazan je na slici 3.9. Svaka boja predstavlja zaseban položaj filtera tijekom sažimanja.

Za razliku od konvolucijskog sloja koji obavlja transformaciju konvolucijskim kernelom, u ovome sloju se sve operacije obavljaju pomoću *max pooling* tenzora [71].



Slika 3.9. Uloga sloja sažimanja u modelu konvolucijskih neuronskih mreža [72]

3.2.2.6. Potpuno povezani sloj

Potpuno povezani sloj (engl. *fully connected layer*) posljednji je sloj pri implementaciji konvolucijskih neuronskih mreža. Glavna funkcija ovog sloja je klasifikacija značajki koje su izdvojene i sažete nizom konvolucijskih slojeva i slojeva sažimanja [64]. Svoj naziv potkrepljuje činjenicom da je svaki ulaz povezan sa svakim izlazom mreže, a koji zasebno čine vektor. Djeluju poput tradicionalne neuronske mreže sadržavajući približno 90% parametara modela. Ulaz potpuno povezanog sloja je posljednji sloj sažimanja, a izlaz mu je N-dimenzionalni vektor u kojem N predstavlja broj klasa prema kojima se obavlja klasifikacija na testnom skupu podataka [57].

Nakon potpuno povezanog sloja, moguće je implementirati softmax funkciju s ciljem prevođenja brojčanih vrijednosti u distribuciju vjerojatnosti. Ovakvu primjenu moguće je pronaći u VGGNet modelu. Na izlazu iz mreže tako je moguće prikazati s kolikom se točnošću može klasificirati pojedini primjerak iz testnog skupa baze podataka. Nadalje, izlaz mreže je moguće primijeniti u drugim algoritmima strojnog učenja bez potrebe za normalizacijom [73].

3.3. Neke popularne mrežne arhitekture

U nastavku je opisana struktura te prednosti i mane nekih od najpoznatijih mrežnih arhitekture dubokog strojnog učenja. Navedene arhitekture prerađene su u eksperimentalnom dijelu rada.

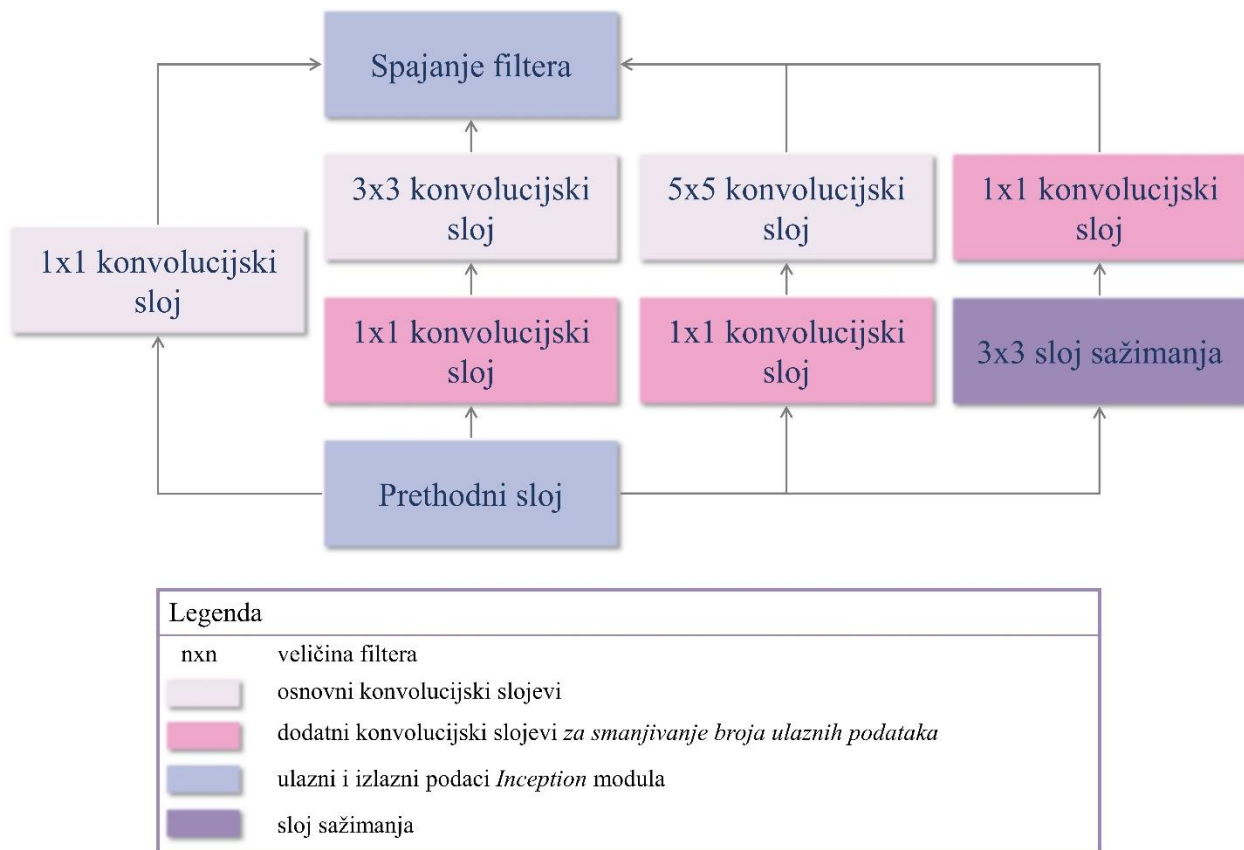
3.3.1. Opis GoogLeNet mreže i prikaz arhitekture

Ovaj model kombinira svojstva ponavljajućih blokovskih paradigmi te strukturu mreže u mreži (engl. *NiN blocks*) koja na kanalima implementira višeslojni perceptron, zasebno za svaki piksel. Blok mreže u mreži sastoji se od konvolucijskog sloja čija se veličina filtera postavlja ovisno o svojstvima mreže kojeg slijede još dva konvolucijska sloja s veličinom filtera 1x1 koji predstavljaju potpuno povezani sloj po pikselu s *ReLU* aktivacijskom funkcijom. Za potpuno izbjegavanje guste povezanosti mreže, *NiN* model koristi *NiN* blokovsku strukturu s jednakim brojem izlaznih kanala i označenih klasa koje slijedi globalni sloj sažimanja umjesto potpuno povezanog sloja pri čemu se sprječava pretjerano prilagođavanje podataka. Najznačajnija prednost *NiN* dizajna je značajno smanjenje broja parametara potrebnih za treniranje modela iako je u praksi ono i dalje dugotrajno [74, 75].

Sastavljena od 22 sloja čineći 9 Inception modula, GoogLeNet arhitekturu uspoređujući s AlexNet modelom određuje približno 12 puta manje parametara uz ostvarenje preciznijih i optimiziranijih rezultata [76]. Pri smanjenju veličine mreže, važnu ulogu ima sloj sažimanja s veličinom filtera 3x3. Strukturnu sličnost osim s AlexNet modelom, dijeli s LeNet i VGGNet modelima [74].

Kao najvažniji dio GoogLeNet modela, Inception modul produbljuje mrežnu strukturu ne povećavajući pri tome računalnu složenost. Modul paralelno izvodi višestruke operacije konvolucije izdvajajući značajke na određenim razinama pomoću filtera različitih veličina. Na slici 3.10. prikazana je arhitektura Inception v1 modela čiji se naziv često koristi kao sinonim za GoogLeNet model predstavljajući temeljnu strukturnu jedinicu ove mreže. Inception blok sastoji se od 4 paralelne razine, a uobičajeni parametar svakog bloka je broj izlaznih kanala po sloju. Početne tri razine koriste konvolucijski sloj s veličinama filtera 1x1, 3x3 te 5x5 za izdvajanje značajki. Na središnjim razinama obavljaju se operacije konvolucije na ulaznim podacima (ili izlaznim podacima iz prethodnog sloja) s filterima veličine 1x1 smanjujući broj ulaznih kanala. Takvi slojevi mrežne arhitekture klasificirani su kao dodatni konvolucijski slojevi. Posljednju razinu čine sloj sažimanja s veličinom filtera 3x3 te konvolucijski sloj s veličinom filtera 1x1, a

zadaća im je smanjenje broj kanala (ili broja karti značajki) u mreži u kojoj dolazi do njihovog povećanja u svakom sloju. Rješenje je problema uskog grla (engl. *bottleneck problem*) u području konvolucijskih neuronskih mreža. Potom slijedi postupak spajanja filtera duž kanala stvarajući izlaz iz bloka [74, 75].



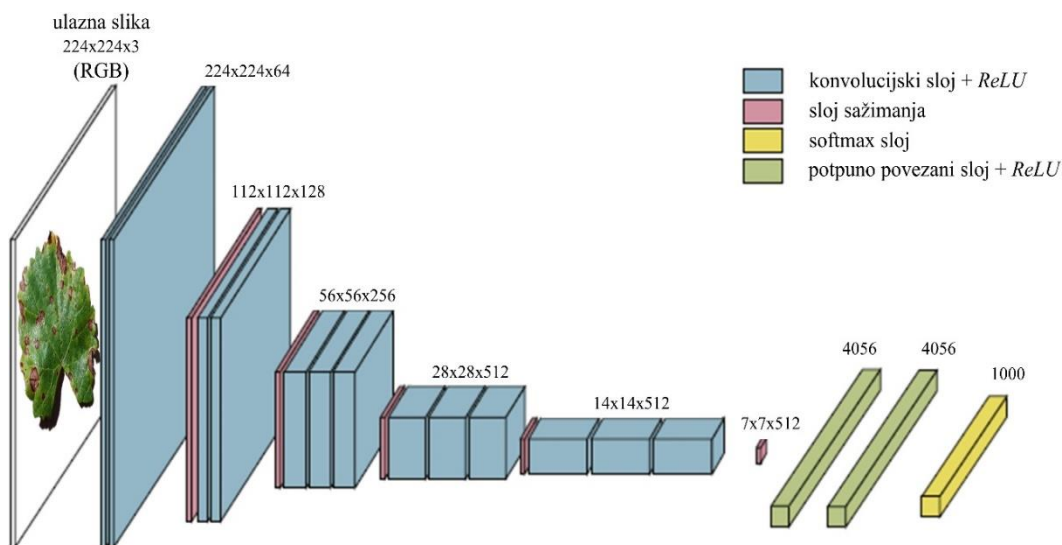
Slika 3.10. Prikaz arhitekture Inception v1 modela [75]

Razvijajući nove verzije dizajna Inception modela poput Inception v3 i Inception v4 modela, pokušava se smanjiti problem nestajućeg gradijenta (engl. *vanishing gradient problem*) [77] koji je česta pojava prilikom produbljanja mreže, a rezultira i pretjeranim prilagođavanjem podataka [54]. Budući da gradijenti kontroliraju proces učenja mreže, moguće je ostvarivanje loših prediktivnih performansi ukoliko je vrijednost gradijenta veoma mala ili jednaka nuli [78]. Nastali problem uspješno se rješava pohranjivanjem optimalnih vrijednosti te zbrajanjem rezultata pomoćnog klasifikatora pomoću algoritma propagacije unatrag. Na kraju procesa učenja, pomoćni klasifikatori se odbacuju, a u fazi ispitivanja njihovi rezultati se ne uračunavaju [54].

3.3.2. Opis VGGNet mreže i prikaz arhitekture

Arhitekturu VGGNet modela čine konvolucijski slojevi s veličinom filtera 3x3 te slojevi sažimanja s veličinom filtera 2x2 između kojih se primjenjuje *ReLU* aktivacijska funkcija. Slijede ih potpuno povezani slojevi određeni s glavninom parametara mreže. Naposljetku, upotrebljava se softmax funkcija koja izračunava vjerojatnosti za svaku klasu [55]. Navedena arhitektura prikazana je na slici 3.11. Glavno svojstvo VGGNet modela je faktoriziranje konvolucijskih filtera, time gradeći duboku mrežu pomoću više manjih slojeva te smanjujući broj parametara za približno 30%. Primjerice, postupkom faktoriziranja konvolucijskih filtera, filter veličine 5x5 moguće je raspodijeliti na dva filtera veličine 3x3 povezanih kaskadno što dovodi do učinkovitijeg izdvajanja karti značajki i smanjenja linearnosti [54]. Uspoređujući ga s GoogLeNet modelom, VGGNet model ima manju dubinu, a izvedba paralelizma u ugrađenim sustavima je značajno kompleksnija. Drastičan nedostatak modela zahtjevno je korištenje memorije zbog ogromnog broja parametara (približno 140 milijuna) što dovodi do usporenja performansi i povećanja vremena treniranja [55, 74].

Pri postupcima optimiziranja, moguće je transformirati operaciju konvolucije u operaciju množenja matrica koja je jedna od najprilagodljivijih operacija na grafičkoj obradbenoj jedinici. Potrebno je pretvoriti trodimenzionalne matrice u dvodimenzionalne, a promatraju se dimenzije matrice značajki poput dubine, širine i dužine svake karte značajki te dimenzije konvolucijske matrice poput izlazne dubine karte značajki, ulazne dubine kanala te jezgrina širina i dužina pri čemu je kernel uvijek kvadratna matrica [55].



Slika 3.11. Prikaz arhitekture VGGNet mreže [64]

3.4. Postupci analize i usporedba performansi

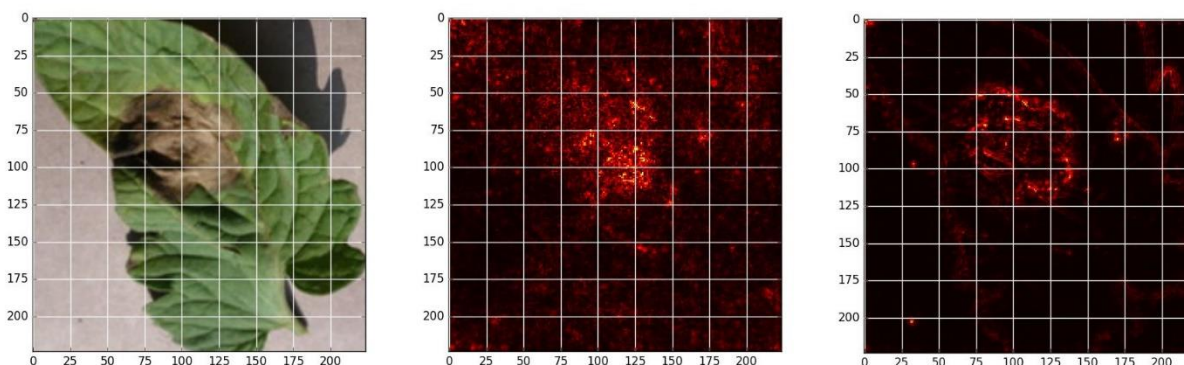
Za postizanje rezultata visoke točnosti, algoritmima dubokog strojnog učenja potrebna je velika količina podataka [79]. Uobičajeni pristup tijekom analiziranja pojedinih CNN arhitektura i mjerenja performansi zahtijeva podjelu baze podataka na trening i testni skup podataka, najčešće u omjeru 80 : 20. Točnost je moguće odrediti poznavajući ishode za podatke u testnom skupu podataka [80]. Unatoč uspješnim rezultatima klasifikatora dubokog strojnog učenja, prikaz je nerazumljiv, stoga oni zahtijevaju specijalizirane algoritme vizualizacije koji lokaliziraju i prepoznaju simptome bolesti kao i zaražene dijelove lista te ih prikazuju [31].

3.4.1. Vizualizacija na primjeru *saliency map*

Saliency map analitička je metoda koja omogućava procjenu važnosti svakog od piksela koristeći jedan prolaz naprijed te jedan prolaz natrag kroz mrežu. Ako vrijednost gradijenta u pikselu ima veliku apsolutnu vrijednost tada ostvaruje veću važnost od onoga s nižom vrijednošću gradijenta, a koji tada ne utječe na izlaznu vrijednost. Izračun *saliency map* prikazan je sljedećom formulom, a prikaz rezultata na listu biljke prikazan je na slici 3.12.

$$G(x) = \frac{df(x)_y}{dx} \quad (3 - 11)$$

Izračunava se gradijent $G(x)$ radi procjene važnosti piksela za ulaznu sliku x . $G(x)$ je tenzor s istim dimenzijama kao i ulazna slika. Pri prolazu naprijed izračunava se izlaz mreže označen s $f(x)$. Potom, prolaskom natrag izračunava se gradijent čvora klase $f(x)_y$ [31].



Slika 3.12. *Saliency map* na primjeru bolesne jedinke [31]

3.4.2. Matrica zabune

Matrica zabune kvadratna je matrica u kojoj svaki redak predstavlja stvarnu klasu koja može biti identificirana ili ne, dok stupci predstavljaju klase koje je odredio klasifikator. Moguće ju je zapisati i u obratnom obliku pri čemu se ne mijenja iznos izračunatih mjera. Točnost modela može se utvrditi promatrajući vrijednosti na glavnoj dijagonali, a pri tome se određuje broj ispravnih klasifikacija. Precizan model s najboljim performansama imat će visoke vrijednosti duž glavne dijagonale, a niske vrijednosti izvan nje [81]. Primjer matrice zabune s općenitim podacima prikazan je na slici 3.13. dok su u tablici 3.1. prikazani njezini mogući slučajevi.

Tablica 3.1. Popis slučajeva u matrici zabune [82]

	Vrijednost klase	Identifikacija algoritma
TP (engl. <i>True Positives</i>)	1	1
FN (engl. <i>False Negatives</i>)	1	0
FP (engl. <i>False Positives</i>)	0	1
TN (engl. <i>True Negatives</i>)	0	0

	Dodijeljena klasa IDENTIFICIRANO	Dodijeljena klasa NEIDENTIFICIRANO
Stvarna klasa IDENTIFICIRANO	TP	FN
Stvarna klasa NEIDENTIFICIRANO	FP	TN

Slika 3.13. Matrica zabune [82]

Ovakva analiza modela korisna je za prepoznavanje slučajeva u kojima su ostvarene visoke točnosti, a da pri tome model pogrešno klasificira iste podatke. Prema pogrešno klasificiranim podacima, potrebno je ponovno podešavanje parametara kako bi ukupna performansa modela bila bolja [81].

3.4.3. Preciznost, odziv, mjera F1 i točnost

Preciznost (engl. *precision*) ili pozitivna prediktivna vrijednost odnosi se na udio relevantnih instanci među ukupnim brojem dohvaćenih instanci. Rezultat preciznosti pokazuje koliko točno klasifikator predviđa da je dodijeljena klasa zapravo ona stvarna [83].

$$P = \frac{TP}{TP + FP} \quad (3 - 12)$$

Odziv (engl. *recall*) je mjera koja se odnosi na udio relevantnih instanci koje su pronađene u ukupnom broju relevantnih instanci. Označava s kolikom točnošću model prepoznaje promatranu klasu [83].

$$R = \frac{TP}{TP + FN} \quad (3 - 13)$$

Mjera F1 (engl. *F1 score*) prikazuje rezultat točnosti testa. Definira se kao harmonijska sredina preciznosti i odziva. Najvišu vrijednost koju ova mjera može postići iznosi 1, a označava najbolju preciznost i odziv. Suprotno, najmanja vrijednost ove mjere iznosi 0, a označava najlošiju preciznost i odziv modela [84].

$$F1 = 2 * \frac{P * R}{P + R} \quad (3 - 14)$$

Točnost (engl. *accuracy*) je mjera koja izražava koliko je predikcija bilo točno, a izražava se u postocima. Omjer je broja točnih predviđanja i ukupnog broja ulaznih uzoraka [85].

$$A = \frac{TP + TN}{TP + FP + TN + FN} \quad (3 - 15)$$

4. EKSPERIMENTALNA ANALIZA I USPOREDBA PERFORMANSI KLASIFIKATORA NA PROBLEMU KLASIFIKACIJE BOLESTI LISTA BILJKE

Svaki postojeći sustav ima jasno definiran skup zahtjeva kojeg je neophodno ispuniti u cilju ostvarenja rezultata visoke točnosti. Ako neki od zahtjeva nije ispunjen, razmatrani sustav prilikom otkrivanja bolesti biljke može pogriješiti, a time dovesti do netočne dijagnoze, liječenja te mogućnosti širenja bolesti na zdrave usjeve [86]. Duboka konvolucijska neuronska mreža dokazavši visoku točnost i performanse pri treniranju modela na raznim skupovima podataka, postala je vodeća metoda za problem klasifikacije. Povećanjem broja slojeva, povećava se i razina učinkovitosti ovog modela [87] koji značajno ovisi o kvaliteti dostupnih podataka, njihovom broju i izdvojenim značajkama [86].

Ovo poglavlje donosi eksperimentalnu analizu i usporedbu rezultata za pet različitih modela klasifikacije temeljenih na metodi duboke konvolucijske neuronske mreže. Prethodno su razjašnjeni svi zahtjevi na sustav, a opisane su i korištene tehnologije. Nakon postavki, provedbe i analize rezultata eksperimenta, predložene su mogućnosti poboljšanja uz osvrt.

4.1. Zahtjevi na sustav

Treniranje modela s većim brojem parametara i slojeva predstavlja izazov u procesima obrade za središnju obradbenu jedinicu koja ne može efikasno iskoristiti paralelizam za veliki broj istodobnih operacija. Zbog pogodne arhitekture i mogućnosti paralelnog izvođenja operacija na dvorazinskoj hijerarhiji procesa i niti, grafička obradbeno jedinica pruža učinkovito rješenje za ovaj problem. Takvi procesi modeliraju se pomoću NVIDIA arhitekture grafičke obradbene jedinice koja se temelji na CUDA (engl. *Compute Unified Device Architecture*) paralelnoj računalnoj platformi, ujedno i modelu aplikacijskog programskog sučelja. Model CUDA stvara heterogeni sustav za obavljanje složenih radnji u kojemu istovremeno sudjeluju obje obradbene jedinice. Točnije, središnja obradbeno jedinica obavlja logičke i matematičke operacije, dok grafička jedinica omogućava paralelizam na razini niti [88, 89]. Biblioteka cuDNN temeljena na modelu CUDA namijenjena je izvođenju i treniranju dubokih neuronskih mreža podržavajući matematičke operacije poput konvolucije unaprijed i unatrag te operacija koje se odvijaju u sloju sažimanja, aktivacijske funkcije i normalizaciju. Programska biblioteka TensorFlow koja je korištena u

ovome radu, oslanja se na visoke performanse grafičke obradbene jedinice i pripadajućih biblioteka [90, 91].

4.2. Korištene tehnologije

Napretkom računalnih tehnologija, omogućena je implementacija velikog skupa neurona stvarajući sve složenije neuronske mreže [92]. Za efikasnu obradu modela potrebne su specijalizirane sklopovske i algoritamske optimizacijske tehnike. U nastavku su prikazane i opisane korištene tehnologije u ovome radu potrebne za programiranje modela dubokog strojnog učenja.

4.2.1. Python i Anaconda

Jedan od razloga zbog kojih je objektno-orijentirani programski jezik Python najkorišteniji za izradu modela i razvoj dubokog strojnog učenja je širok raspon odabira biblioteka koje sadrže prethodno implementirane module. Osim što pripada programskim jezicima više razine, jednostavne je sintakse i opsežno dokumentiran. Također, moguće ga je kombinirati s drugim programskim jezicima za učinkovito rješavanje problema. Uz objektno-orijentirani, podržava i proceduralni te funkcionalni stil programiranja. Za potrebe prikaza podataka te usporedbe učinkovitosti pojedinih modela, Python omogućava korištenje niza alata za vizualizaciju [93]. U svrhu praktičnijeg i preglednijeg rada s paketima koda, u ovom radu korištena je Anaconda – distribucija Pythona i R programskog jezika otvorenog koda specijalizirana za strojno učenje, prediktivnu analitiku i podatkovne znanosti. Upravljenje paketima ostvareno je kroz Anacondin vlastiti sustav *conda* koji je izdvojen kao zaseban projekt otvorenog koda [94].

4.2.2. MATLAB

Programski jezik MATLAB u ovom radu korišten je pri implementaciji prerađenog algoritma koji se temelji na metodi k-najbližih susjeda za segmentaciju slika bolesnih listova biljaka s ciljem odvajanja i prepoznavanja bolesnih dijelova lista te određivanja ukupne površine lista zahvaćene bolešću. Ovaj programski jezik sve računske operacije svodi na operacije s matricama. Često je korišten alat pri analiziranju podataka, razvoju algoritama za strojno učenje i robotski vid te pri kreiranju matematičkih modela. Svoju primjenu pronalazi i u obradi signala, segmentaciji slika i robotici. Podržava paradigme objektno-orijentiranog programiranja, C programski jezik, Javu i SQL. Omogućava izradu 3D grafova te grafičkog korisničkog sučelja [95].

4.2.3. TensorFlow i Keras

TensorFlow je programska biblioteka otvorenog koda korištena za stvaranje modela strojnog učenja [96]. Kako bi ih izradila i trenirala, ova biblioteka promatra graf protoka podataka u kojemu čvorovi (engl. *actors*) predstavljaju operacije koje se izvode na podacima čiji je tok modeliran usmjerenim granama. Kako se skup odabranih čvorova može izvoditi istovremeno ili slijedno bilo kojim redoslijedom, omogućujući paralelizam za implementaciju modela, razumljivo je najkorištenije rješenje za stvaranje neuronskih mreža s više slojeva, klasifikaciju, predikciju i modele klasičnog strojnog učenja [97]. U višenitnim procesima na razini poslužitelja dokazano ostvaruje najbolje rezultate. Dodatno, za mobilne uređaje dizajniran je *TensorFlow Lite* koji pruža aplikacijsko programsko sučelje za izvođenje računskih operacija strojnog učenja na Android uređajima [98]. Svoje ime temelji na pojmu tenzor, matematičkoj veličini određenoj s više od tri broja uz dodatna svojstva. Po zakonu transformacije tenzora, pri prelasku iz jednog koordinatnog sustava u drugi, mijenjaju se i komponente tenzora [98]. Pisan u Pythonu, razvojni okvir (engl. *framework*) za duboko strojno učenje Keras izvodi se na najvišoj programskoj razini *TensorFlow*-a. Sadržava brojne implementacije slojeva neuronskim mreža, aktivacijskih i optimizacijskih funkcija. Treniranje modela dubokog strojnog učenja izvodi na grafičkoj obradbenoj jedinici ili tenzorskim obradbenim jedinicama najčešće u kombinaciji s CUDA razvojnom okolinom [99].

4.2.4. NumPy i SciPy

Kao osnovna biblioteka u Pythonu za računanje u znanstvene svrhe, NumPy olakšava rad s matricama i višedimenzionalnim nizovima, obavlja trigonometrijske, statističke i algebarske operacije na nizovima, sadržava generator slučajnih brojeva te transformacijske funkcije od kojih je najkorištenija Fourierova transformacija [100, 101]. Također, omogućava integraciju koda u C/C++ programskim jezicima pomoću odgovarajućih alata. Koristeći NumPy nizove kao osnovnu strukturu podataka, SciPy sadržava module za optimizaciju, interpolaciju, obradu signala te linearnu algebru [101, 102].

4.2.5. OpenCV i Matplotlib

OpenCv je biblioteka otvorenog koda za stvaranje modela strojnog učenja i računalnog vida. Ova biblioteka sadržava više od 2500 optimiziranih algoritama koji se primjenjuju u detekciji i prepoznavanju lica, identifikaciji i praćenju objekata, stvaranju oblaka 3D točaka te proširene

stvarnosti. Uz Python, podržava i C++, Javu te MATLAB programske jezike [103]. Za vizualizaciju podataka, u ovome radu korištena je Matplotlib biblioteka pisana u Pythonu. Omogućava objektno-orijentirano aplikacijsko programsko sučelje pomoću kojeg se izrađuju grafovi za opću namjenu [104].

4.3. Postavke eksperimenta

U eksperimentalnom dijelu rada prikazan je postupak klasifikacije bolesti lista biljke na 5 različitih arhitektura dubokog strojnog učenja. Sve konvolucijske neuronske mreže trenirane su i testirane na odabranom skupu podataka koji uključuje bolesne i zdrave jedinice kukuruza podijeljene u 4 klase. Promatrane su bolesti: siva pjegavost lista, obična hrđa te palež lista. Korištena baza podataka dio je *New Plant Diseases Dataset*-a opisane u teorijskom dijelu rada. Prilikom višestrukog treniranja klasifikatora, primjećena je izuzetno velika točnost u prepoznavanju jedinice oboljele od paleži lista. Zbog specifičnosti simptoma navedene bolesti i velikog broja trening primjera, a kako bi se izbjeglo pretjerano prilagođavanje podataka trening skupu, broj podataka klase smanjen je slučajnim odabirom za 40%. Trening i test skup podataka podijeljeni su u omjeru 80:20 u ukupnom broju 5809 podataka za treniranje i 1829 podataka za testiranje. Sve slike su u boji te su predobradom dimenzionirane u veličinu 224x224x3 piksela.

Kako bi se omogućila ravnopravna usporedba rezultata svih eksperimenata, hiperparametri su postavljeni na sljedeće vrijednosti:

- stopa učenja: odabrana automatizmom algoritma prilikom treniranja mreže pomoću *ReduceLROnPlateau*
- broj epoha: 20
- *batch size*: 8
- optimizacijski algoritmi: Adadelta, Adam

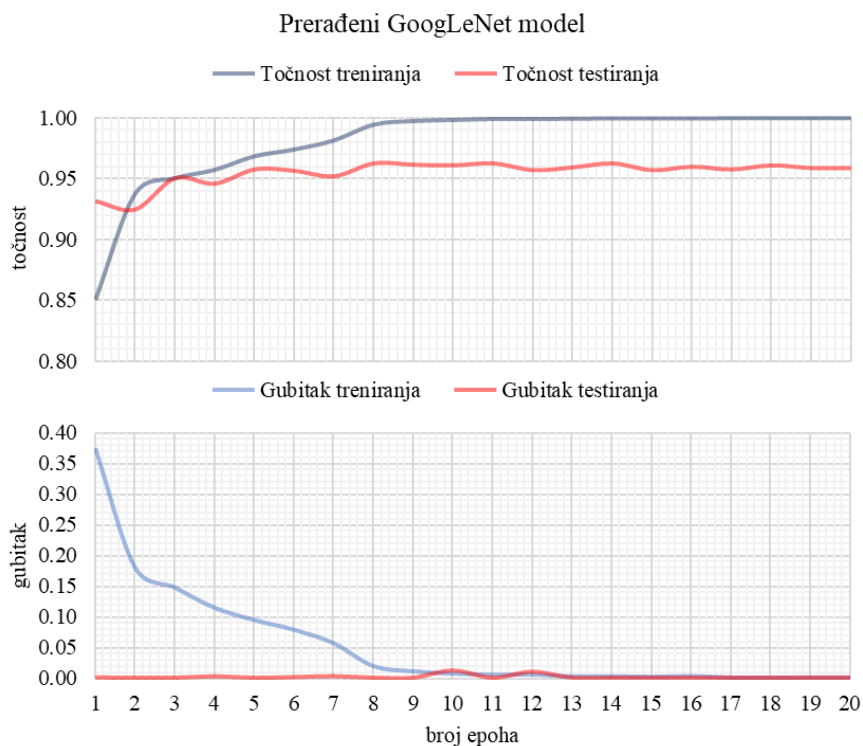
Pri provedbi eksperimenata korištena je središnja obradbeno jedinica Intel Core i5, 12 GB radne memorije te grafička obradbeno jedinica NVIDIA GeForce GTX 1050 s 4 GB memorije na Windows 10 operacijskom sustavu. Implementacija mreža i predobrada podataka provedena je u

Anaconda3 distribuciji s Python verzijom 3.6 te Keras-GPU i TensorFlow-GPU paketom. U prosjeku, svaki model se trenirao 3 i pol sata.

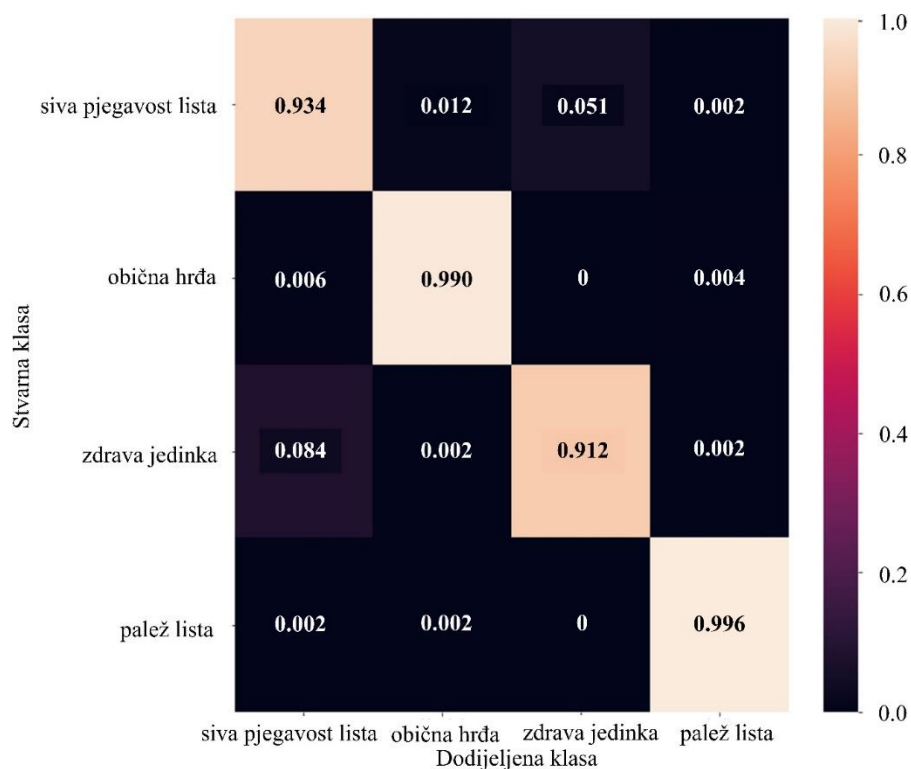
4.4. Provedba eksperimenta

Eksperimentalna analiza raspoznavanja bolesti na listu biljke provedena je na prerađenim ili prethodno treniranim konvolucijskim neuronskim mrežama uz standardizirane vrijednosti hiperparametara. Prethodno trenirani modeli sadržavaju težine naučene treniranjem na javno dostupnoj ImageNet bazi slika u kojoj je svaki čvor hijerarhije prikazan stotinama i tisućama slika te su potom trenirani na odabranom skupu slika zdravih i bolesnih jedinki kukuruza. Pristupom prijenosnog učenja, iskorištene su već naučene karte značajki. Različito od njih, prerađeni modeli trenirani su jedino na odabranim jedinkama iz dijela *New Plant Diseases Dataset-a*.

Na slici 4.1. te slici 4.2. prikazani su rezultati klasificiranja bolesti biljke pomoću prerađenog GoogLeNet modela.



Slika 4.1. Prikaz rezultata analize za prerađeni GoogLeNet model

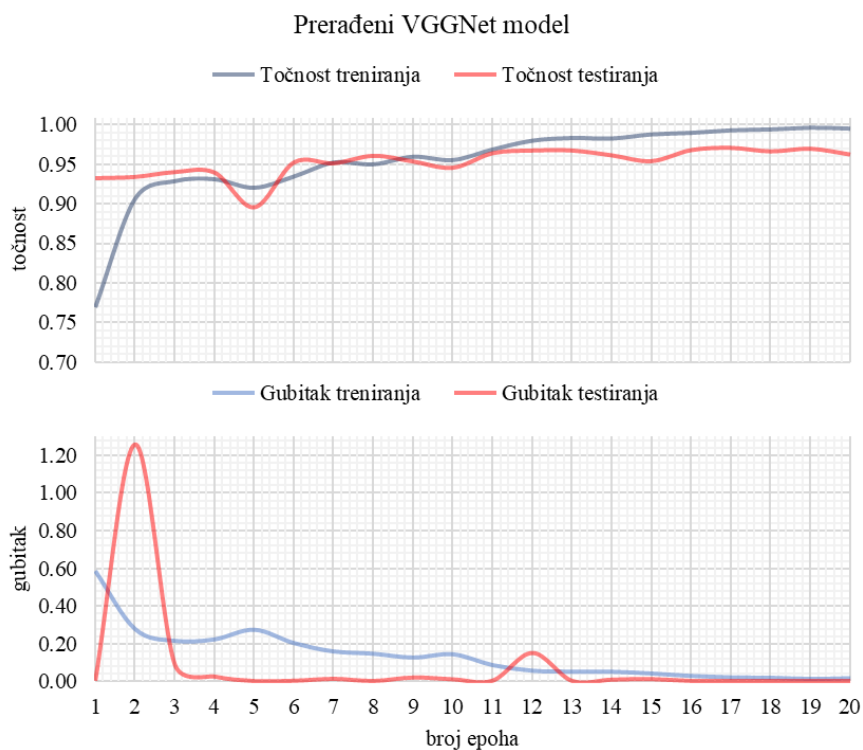


Slika 4.2. Matrica zabune za prerađeni GoogLeNet model

Sama implementacija modela odrađena je po uzoru na Inception modul čija je struktura opisana u teorijskom dijelu rada, poglavlju 3.3.1. U svrhu ostvarivanja veće točnosti modela i korištenja novih dostignuća u tehnologiji, klasična *ReLU* aktivacijska funkcija u svim konvolucijskim slojevima modela, zamijenjena je Mish aktivacijskom funkcijom. Korištenje Mish aktivacijske funkcije nije namijenjeno prilikom postupka klasifikacije u potpuno povezanom sloju modela jer izračunom logaritma može doći do negativnog rezultata za kojeg logaritamska funkcija nije definirana i tako rezultirati NaN vrijednošću (*Not a Number*).

Iako je točnost testiranja određena manjom vrijednošću od one treniranja, nije došlo do pretjeranog prilagođavanja podataka jer je gubitak testiranja (engl. *testing loss*) manji od gubitka treniranja (engl. *training loss*), a analizom vrijednosti u matrici zabune ne dolazi do njihove značajne varijacije.

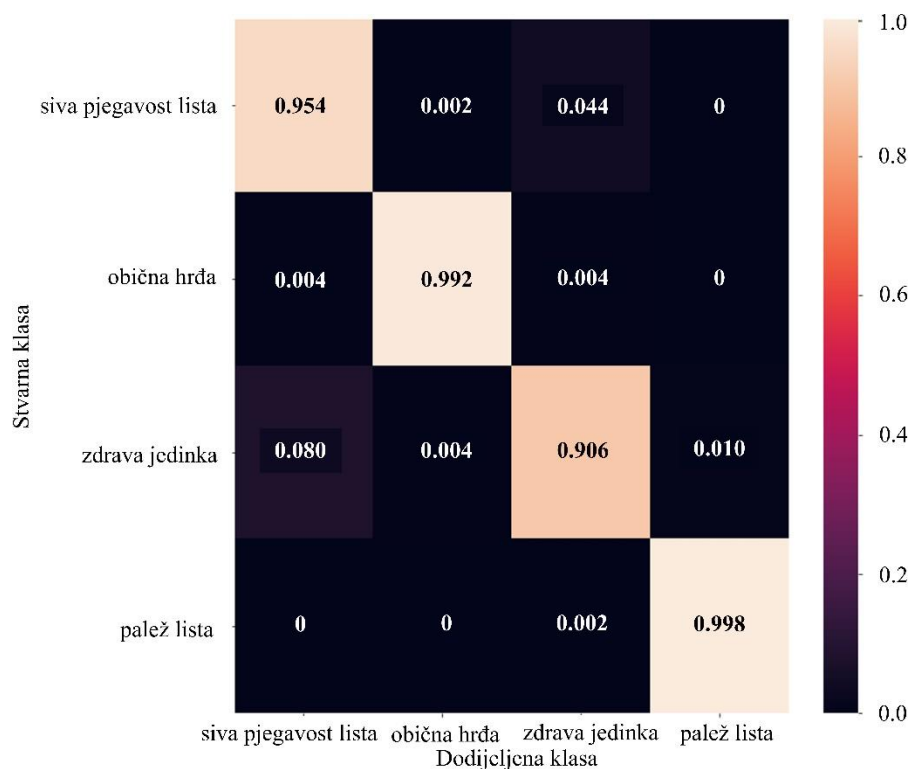
Sljedeći model izrađen je po uzoru na inačice VGGNet modela poput VGG13, VGG16 i VGG19, no s manjim brojem slojeva zbog izuzetno velike sklopovske zahtjevnosti modela i velikog broja potrebnih parametara. Također, u konvolucijskim slojevima korištena je *ReLU* aktivacijska funkcija radi jednostavnosti i manjeg broja izvršenih operacija. Od navedenih modela, prerađeni VGGNet model ostvario je najduže vrijeme treniranja.



Slika 4.3. Prikaz rezultata analize za prerađeni VGGNet model

Pojedine grube greške gubitka testiranja u drugoj epohi prisutne su u analizi prerađenog VGGNet modela uz prikaz ostalih rezultata na slikama 4.3. i 4.4. U ostalim epohama, vrijednosti gubitka testiranja manje su od vrijednosti gubitka treniranja, uz iznimku u dvanaestoj epohi. U VGGNet modelu također je primjećena veća točnost za razliku od, prethodno opisanog, prerađenog GoogLeNet modela.

U matrici zabune najtočnije se klasificira bolest palež lista dok je najmanja točnost primijećena kod klasifikacije zdrave jedinke kukuruza.

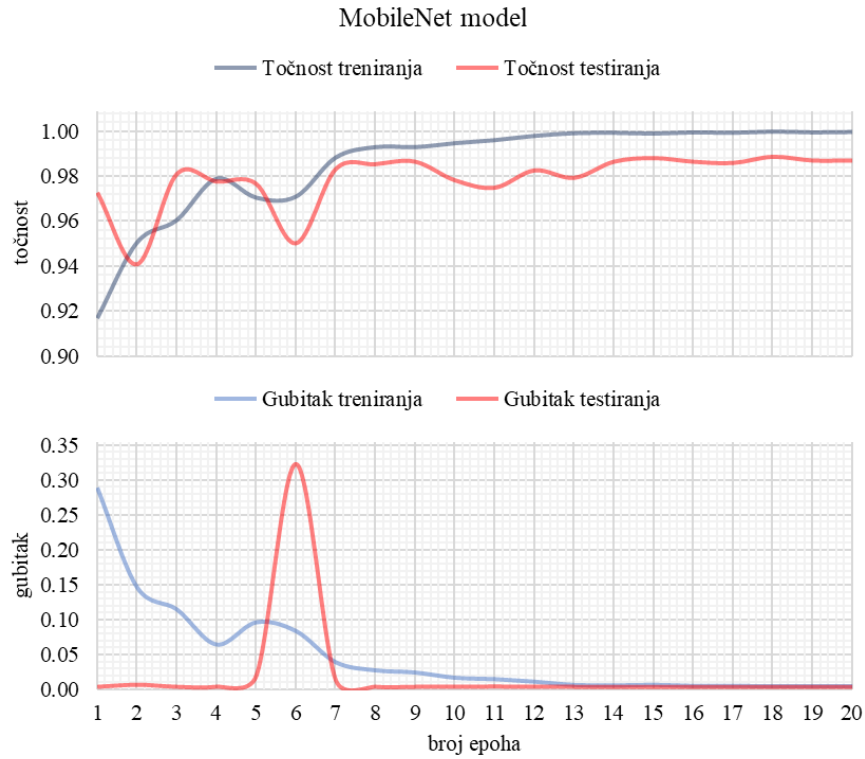


Slika 4.4. Matrica zabune za prerađeni VGGNet model

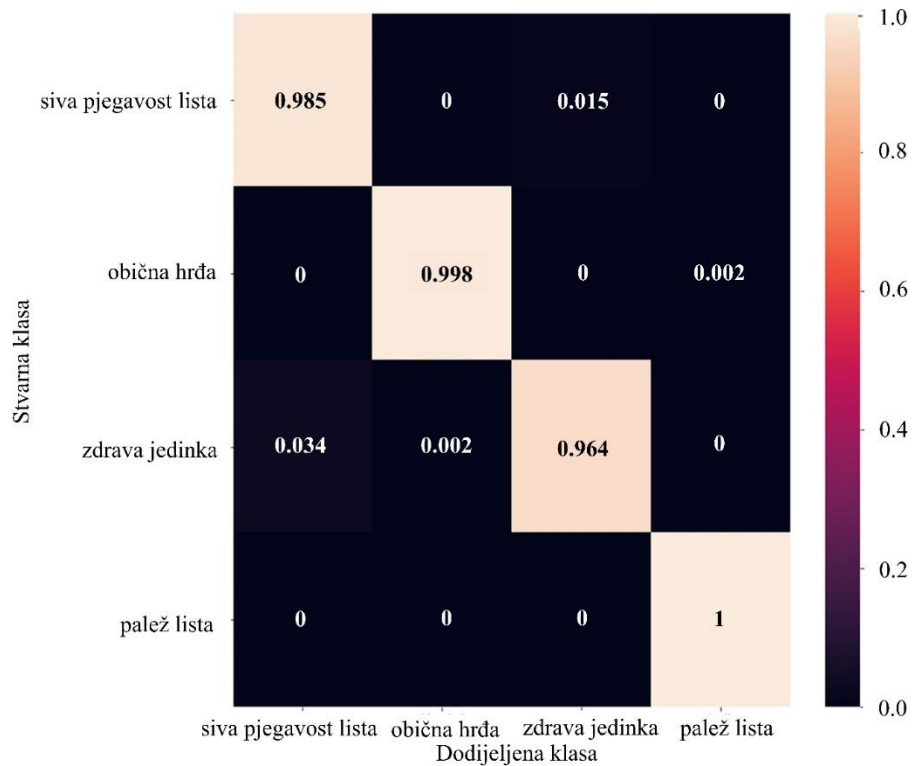
Kako bi se olakšalo prepoznavanje i praćenje biljnih bolesti stručnjacima na terenu, pretežno je važna implementacija modela dubokog strojnog učenja na mobilne uređaje pa je u tu svrhu korištena MobileNet arhitektura. Strukturom i brojem parametara je najmanja od ostalih (sadrži nešto više od 4 milijuna parametara), treniranje je brže i posljedično joj je potrebno najmanje vremena za učenje parametara. Taj postupak dodatno olakšava činjenica da je korišten prethodno trenirani model uz tehnike prijenosnog strojnog učenja.

Rezultati treniranja i testiranja prikazani su na slikama 4.5. i 4.6. I u ovom modelu primijećena je gruba pogreška gubitka testiranja u šestoj epohi, no u ostalima su njene vrijednosti manje od vrijednosti gubitka treniranja. Točnosti treniranja i testiranja postižu veće vrijednosti uspoređujući s prerađenim GoogLeNet i VGGNet modelima.

U matrici zabune, klasifikacija bolesti palež lista ostvaruje vrijednosti 1. Točnije, klasifikator je sve primjerke bolesnih jedinki paleži lista iz testnog skupa baze podataka prepoznao i svrstao upravo u tu klasu. Analizom ostalih metoda i dobivenih rezultata ne dolazi do pretjeranog prilagođavanja podataka, a ovaj problem još je dodatno spriječen smanjenjem baze slika paleži lista za treniranje mreže za 40% zbog specifičnosti simptoma.



Slika 4.5. Prikaz rezultata analize za MobileNet model

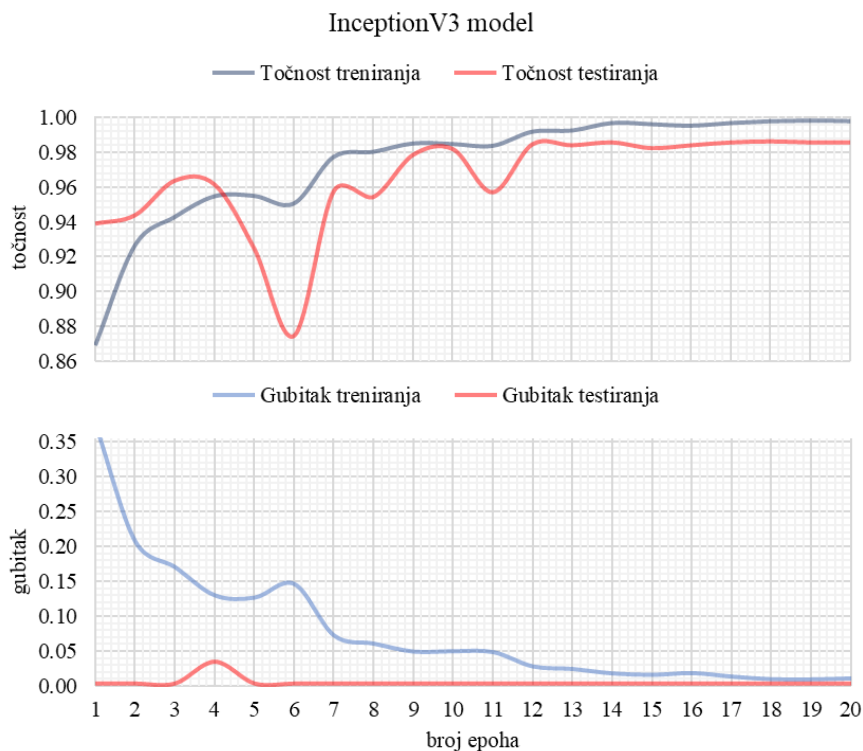


Slika 4.6. Matrica zabune za MobileNet model

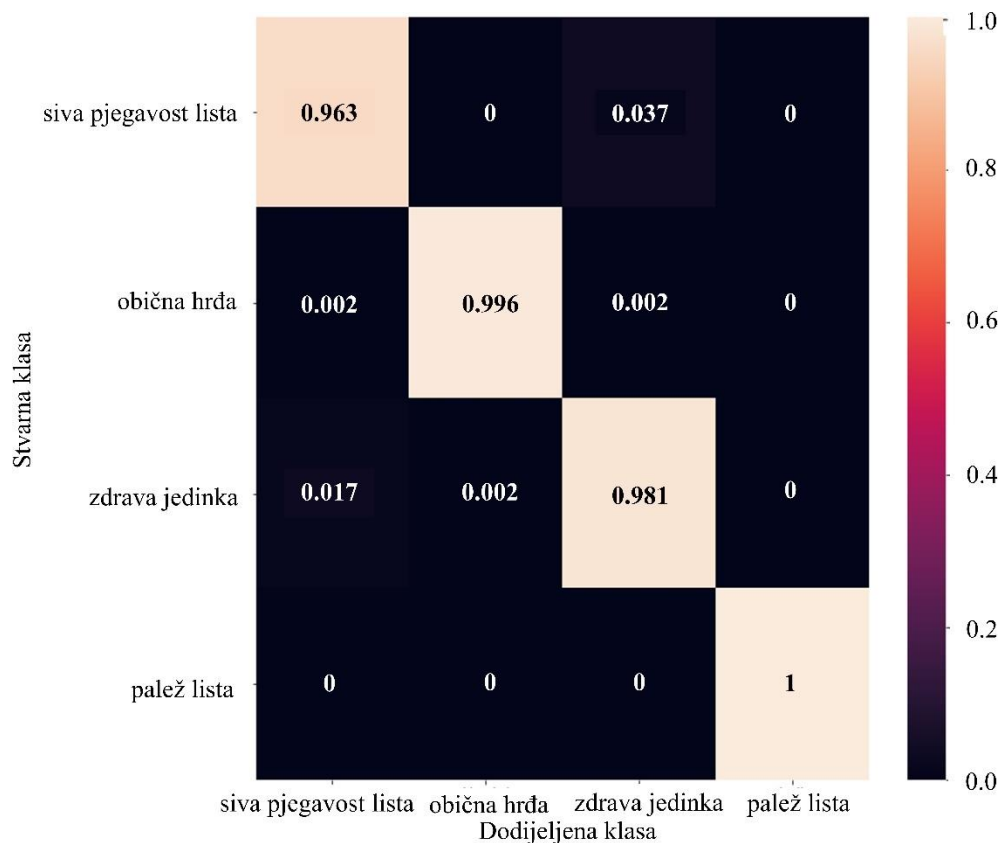
Slijedi InceptionV3 model koji je prethodno treniran na primjercima slika iz ImageNet baze podataka, a potom treniran na primjercima zdravih i bolesnih jedinki kukuruza. U posljednjem sloju konvolucijske neuronske mreže korišten je klasifikator klasičnog strojnog učenja, točnije metoda potpornih vektora koja je podjednako korištena za regresijske i klasifikacijske probleme. Potrebno je koristiti linearnu aktivacijsku funkciju u posljednjem sloju te *hinge loss*, jednu od funkcija za treniranje klasifikatora koja je u ovom slučaju potrebna radi realizacije metode potpornih vektora. Rezultati su prikazani na slikama 4.7 i 4.8.

Spoj metoda klasičnog i dubokog strojnog učenja ostvario je nezamjetno manju točnost u usporedbi s MobileNet modelom iako mrežna arhitektura zahtijeva više parametara. Točnosti treniranja i testiranja imaju slične vrijednosti, a gubitak testiranja je u svim epohama manji od gubitka treniranja.

U matrici zabune klasifikator je ponovo sve primjerke bolesnih jedinki paleži lista iz testnog skupa baze podataka uspješno prepoznao i svrstao ih u klasu jedinki bolesnih od paleži lista, a najmanja točnost primijećena je prilikom klasifikacije jedinki koje boluju od sive pjegavosti lista. Podjednako su određene bolest obična hrđa te zdrave jedinke kukuruza.



Slika 4.7. Prikaz rezultata analize za InceptionV3 model

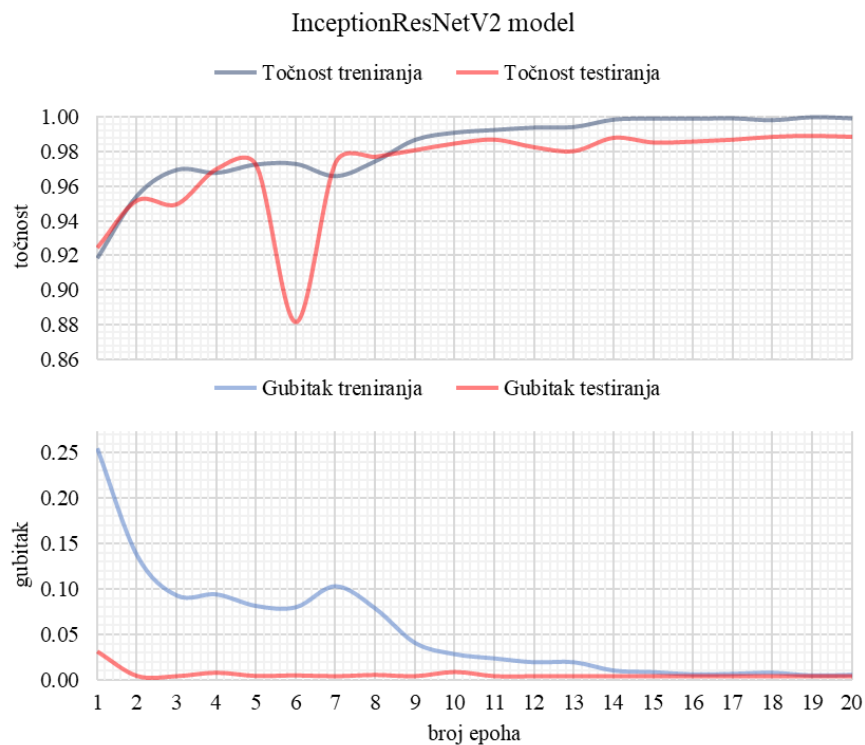


Slika 4.8. Matrica zabune za InceptionV3 model

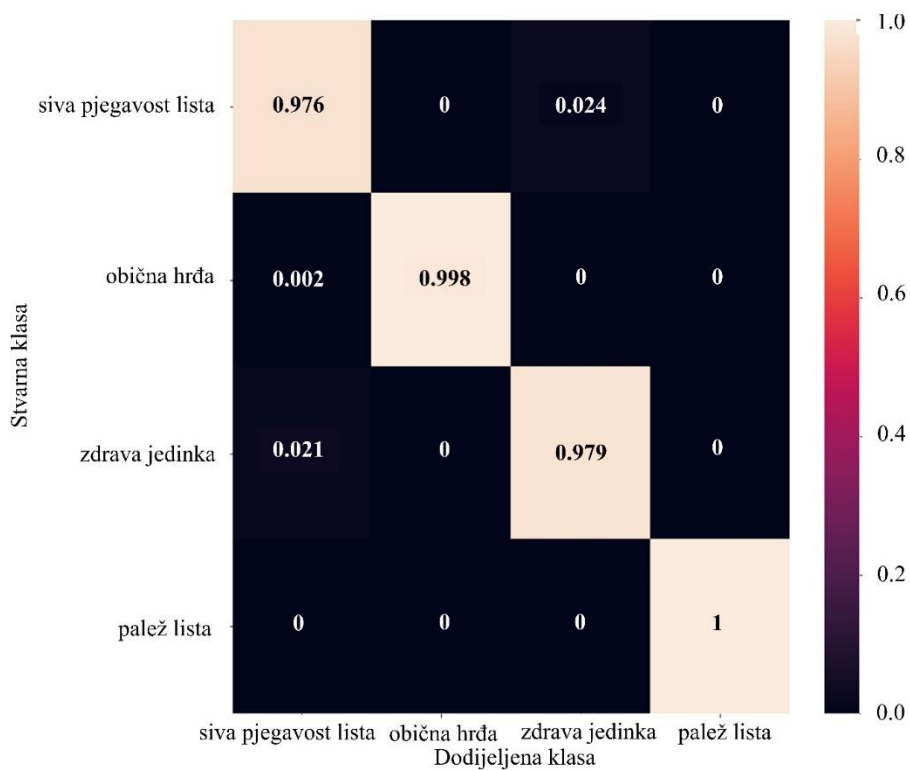
Posljednji trenirani model je InceptionResNetV2 model, a radi se o mrežnoj arhitekturi koja je spoj ResNet te Inception modela s više od 55 milijuna parametara što ju čini najvećom mrežom u usporedbi s korištenim modelima u ovome radu.

Točnosti treniranja i testiranja ostvaruju najveće vrijednosti od ostalih modela, a gubitak treniranja je u svim epohama manji od gubitka treniranja. Na slici 4.9. primijećena je jedna gruba greška u šestoj epohi.

Na slici 4.10. prikazana je matrica zabune za ovaj model, a kao i kod prethodno treniranih modela, klasifikacija bolesti palež lista ostvaruje vrijednost 1, to jest, i ovaj klasifikator sve testne slike jedinki bolesnih od paleži lista uspješno klasificira te svrstava u tu klasu. Ostale vrijednosti klasifikacije pojedine vrste su također veoma velike s manjim brojem pogrešno prepoznanih primjeraka. Primjerice, prilikom testiranja klasifikatora za jedinke bolesne od sive pjegavosti lista, klasifikator ih prepoznaje s točnošću od 96.3%, dok ih u ostalom postotku svrstava u pogrešnu klasu, kao jedinku bolesnu od obične hrđe ili zdravu jedinku.



Slika 4.9. Prikaz rezultata analize za InceptionResNetV2 model



Slika 4.10. Matrica zabune za InceptionResNetV2 model

4.5. Usporedba rezultata

Klasifikacijskim izvještajem na kraju svakog treninga modela, dobivene su vrijednosti preciznosti, odziva te mjere F1 za svaku pojedinu klasu. U tablici 4.1. prikazane su njihove prosječne vrijednosti za svaki model uz točnost.

Tablica 4.1. Usporedba performansi modela

Model	Prosječna preciznost	Prosječni odziv	Prosječna mjera F1	Točnost
Prerađeni GoogLeNet	0.958	0.958	0.958	0.955
Prerađeni VGGNet	0.961	0.962	0.961	0.962
MobileNet	0.986	0.987	0.986	0.987
InceptionV3	0.986	0.985	0.985	0.986
InceptionResNetV2	0.988	0.988	0.988	0.989

Analizirajući navedene mjere, grafove koji prikazuju odnos točnosti treniranja i testiranja kao i njihovih gubitaka te matrice zabune, najbolje performanse i najveću točnost ostvaruje InceptionResNetV2 model dok je najlošiji prerađeni GoogLeNet model. Klasifikatori najtočnije prepoznaju jedinke bolesne od paleži lista (gotovo uvijek sa 100%) čije testne primjere svrstavaju upravo u tu klasu, dok s nešto manjom točnošću prepoznaju jedinke bolesne od sive pjegavosti lista i zdrave jedinke koje često klasificiraju kao jedinke bolesne od obične hrđe zbog sličnosti simptoma ili nedostatka istih.

4.6. Osvrt i mogućnosti poboljšanja

Visoke vrijednosti točnosti omogućuju upotrebu svih modela u svrhe klasifikacije zdravih i bolesnih jedinki kukuruza, a trenirani MobileNet daje mogućnost implementacije u mobilne uređaje. Za mogućnosti poboljšanja prepoznavanja bolesti lista biljke, predloženi modeli mogu se trenirati na većem skupu podataka uz odgovarajuću sklopovsku podršku. Na taj način bi se proširio opseg upotrebe dubokog strojnog učenja u navedene svrhe. Također, skup podataka u ovome radu trebao bi se proširiti s novim primjerima jedinki iz prethodno klasificiranih vrsta biljaka te obogatiti novim. Treniranje na raznolikijem skupu podataka značajno bi doprinuo široj upotrebljivosti i pomoći stručnjacima. U same modele moglo bi se implementirati više konvolucijskih slojeva za produbljivanje arhitekture uz korištenje suvremenih aktivacijskih funkcija.

5. ZAKLJUČAK

Iako su pred preciznu poljoprivredu i suvremene tehnologije poput dubokog strojnog učenja stavljeni svakodnevni izazovi u svrhu sprječavanja širenja epidemije, boljoj kvaliteti usjeva i njihovu očuvanju, tehnologije i mrežne arhitekture opisane u ovome radu značajno doprinose ostvarenju boljih dijagnostičkih metoda uz naglasak na objektivnu pomoć stručnjacima u donošenju odluka. Koristeći sve prednosti dubokog strojnog učenja nad metodama klasičnog, stvorena je mogućnost izdvajanja značajki bez upotrebe metoda segmentacije koja dokazano često smanjuje učinkovitost treniranja mreže uklanjajući značajke iz signala. Zbog toga se konvolucijske neuronske mreže smatraju najperspektivnijim pristupom raspoznavanju bolesti lista biljke i klasifikaciji uopće. Nadalje, novijim pristupom poput korištenja prijenosnog učenja smanjena je potreba za velikim i raznolikim skupom podataka kojeg je uobičajeno veoma teško prikupiti.

Cilj i važnost ovoga rada je ostvariti poboljšanje i brži razvitak dijagnostičkih metoda te istražiti važnost i uspješnost novih tehnologija u području računarskih znanosti. Preciznost i visoke točnosti implementiranih modela omogućuju upotrebu svih modela u klasifikacijske svrhe za zdrave i bolesne jedinke kukuruza, a riječ je o prerađenim modelima GoogLeNet i VGGNet te prethodno treniranim modelima MobileNet, InceptionV3 i InceptionResNetV2. Najbolje performanse ostvario je InceptionResNetV2 model s 98.9% točnosti. Važno je istaknuti MobileNet model koji se zbog svoje strukture, manjeg broja parametara, ali i velike točnosti može lako ugraditi u mobilne uređaje povećavajući raspon upotrebe dubokog strojnog učenja u ostalim granama djelatnosti. Također, spoj metode potpornih vektora koja pripada klasičnom strojnom učenju te konvolucijskih neuronskih mreža iz dubokog strojnog učenja ostvaruje zapaženo dobar rezultat prilikom klasifikacije.

Preostaje istražiti *black box* i povećati broj studija koji se bave problemom vizualizacije te otkriti detaljni tijek samog postupka klasifikacije. Znanje o procesima i načinu računalnog učenja koristeći velik skup podataka, značajno bi doprinuo području znanosti i razotkrio nove postupke pri dijagnostičkim postupcima te oponašanja biološke organizacije i hijerarhije neuronskih mreža.

LITERATURA

- [1] T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, Proceedings of 12th International Conference on Pattern Recognition, sv. 1., str. 582.-585., Jeruzalem, 1994.
- [2] S. P. Mohanty, D. P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Frontiers in plant science, sv. 7., str. 1419., rujan 2016.
- [3] D. C. He, J. S. Zhan, L. H. Xie, Problems, challenges and future of plant disease management: from an ecological point of view, Journal of Integrative Agriculture, br. 4., sv. 15, str. 705.-715., travanj 2016.
- [4] A. M. G. J. Hanson, M. G. Joel, A. Joy, J. Francis, Plant leaf disease detection using deep learning and convolutional neural network, International Journal of Engineering Science, sv. 5324, str. 2554.-2558., ožujak 2017.
- [5] S. W. Zhang, Y. J. Shang, L. Wang, Plant disease recognition based on plant leaf image, J. Anim. Plant Sci, br. 3., sv. 25, str. 42.-45., siječanj 2015.
- [6] M. Gašparović, M. Zrinjski, Đ. Barković, D. Radočaj, An automatic method for weed mapping in oat fields based on UAV imagery, Computers and Electronics in Agriculture, sv. 173., str. 105385., lipanj 2020.
- [7] H. Wang, G. Li, Z. Ma, X. Li, Image recognition of plant diseases based on principal component analysis and neural networks, 2012 8th International Conference on Natural Computation, str. 246.-251., Chongqing, 2012.
- [8] H. Wang, G. Li, Z. Ma, X. Li, Application of neural networks to image recognition of plant diseases, 2012 International Conference on Systems and Informatics (ICSAI2012), str. 2159.-2164., Yantai, 2012.
- [9] P. D. Sharma, Plant pathology, Alpha Science International Limited, Oxford, 2006.
- [10] S. D. Ellis, M. J. Boehm, Plants get sick too! An introduction to plant diseases, Agriculture and Natural Resources, Columbus, 2008.

- [11] A. Van Maanen, X. M. Xu, Modelling plant disease epidemics, *European Journal of Plant Pathology*, br. 7., sv. 109, str. 669.-682., rujan 2003.
- [12] D. A. Shah, P. A. Paul, E. D. De Wolf, L. V. Madden, Predicting plant disease epidemics from functionally represented weather series, *Philosophical Transactions of the Royal Society B*, br. 1775., sv. 374, str. 20180273., svibanj 2019.
- [13] P. S. Ojiambo, J. Yuen, F. Van den Bosch, L. V. Madden, Epidemiology: past, present, and future impacts on understanding disease dynamics and improving plant disease management—a summary of focus issue articles, *Phytopathology*, br. 10., sv. 107, str. 1092.-1094., rujan 2017.
- [14] K. A. Garrett, M. Nita, E. D. De Wolf, P. D. Esker, L. Gomez-Montano, A. H. Sparks, *Plant pathogens as indicators of climate change*, *Climate change*, Elsevier, Amsterdam, Oxford, Waltham, 2016.
- [15] P. R. Birch, A. O. Avrova, A. Dellagi, C. Lacomme, S. S. Cruz, G. D. Lyon, Programmed cell death in plants in response to pathogen attack, *Annual Plant Reviews online*, str. 184.-208., travanj 2018.
- [16] Powdery mildew, Wikiwand, dostupno na https://www.wikiwand.com/en/Powdery_mildew [21.3.2020.]
- [17] M. B. Riley, M. R. Williamson, O. Maloy, *Plant Disease Diagnosis*, *The Plant Health Instructor*, sv. 10., siječanj 2002.
- [18] S. A. Miller, R. R. Martin, Molecular diagnosis of plant disease, *Annual Review of Phytopathology*, br. 1., sv. 26, str. 409.-432., rujan 1988.
- [19] S. S. Abu-Naser, K. A. Kashkash, M. Fayyad, Developing an expert system for plant disease diagnosis, *Journal of Artificial Intelligence*, br. 2., sv. 1., str. 78.-85., 2008.
- [20] N. Petrellis, A smart phone image processing application for plant disease diagnosis, 2017 6th International Conference on Modern Circuits and Systems Technologies (MOCASST), str. 1-4., Thessaloniki, 2017.

- [21] A. K. Mahlein, Plant disease detection by imaging sensors—parallels and specific demands for precision agriculture and plant phenotyping, *Plant disease*, br. 2., sv. 100, str. 241.-251., siječanj 2016.
- [22] G. Saradhambal, R. Dhivya, S. Latha, R. Rajesh, Plant disease detection and its solution using image classification, *International Journal of Pure and Applied Mathematics*, br. 14., sv. 119, str. 879.-884., 2018.
- [23] S. Paul, *Application of Biomedical Engineering in Neuroscience*, Springer, Singapur, 2019.
- [24] S. Bhattarai, New Plant Disease Dataset, kaggle.com, dostupno na: <https://www.kaggle.com/vipooooool/new-plant-diseases-dataset> [2.3.2020.]
- [25] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture*, sv. 145., str. 311.-318., veljača 2018.
- [26] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, D. P. Hughes, Deep learning for image-based cassava disease detection, *Frontiers in plant science*, sv. 8., str. 1852., listopad 2017.
- [27] J. Chen, J. Chen, D. Zhang, Y. Sun, Y. A. Nanekaran, Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture*, sv. 173., str. 105393., siječanj 2020.
- [28] P. Sharma, Y. P. S. Berwal, W. Ghai, Performance analysis of deep learning CNN models for disease detection in plants using image segmentation, *Information Processing in Agriculture*, studeni 2019.
- [29] M. Türkoğlu, D. Hanbay, Plant disease and pest detection using deep learning-based features, *Turkish Journal of Electrical Engineering & Computer Sciences*, br. 3., sv. 27., str. 1636.-1651., svibanj 2019.
- [30] S. H. Lee, H. Goëau, P. Bonnet, A. Joly, New perspectives on plant disease characterization based on deep learning, *Computers and Electronics in Agriculture*, sv. 170., str. 105220., ožujak 2020.

- [31] M. Brahim, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, A. Moussaoui, Deep learning for plant diseases: detection and saliency map visualisation, Human and Machine Learning, str. 93.-117., lipanj 2018.
- [32] Y. Toda, F. Okura, How convolutional neural networks diagnose plant disease, Plant Phenomics, sv. 2019., str. 9237136., svibanj 2019.
- [33] J. G. A. Barbedo, Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification, Computers and electronics in agriculture, sv. 153., str. 46.-53., listopad 2018.
- [34] Bit depth tutorial, Cambridge in colour, dostupno na <https://www.cambridgeincolour.com/tutorials/bit-depth.htm> [23.5.2020.]
- [35] P. Sharma, P. Hans, S. C. Gupta, Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques, 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), str. 480.-484., Noida, 2020.
- [36] P. Bhatt, S. Sarangi, S. Pappula, Comparison of CNN models for application in crop health assessment with participatory sensing, 2017 IEEE Global Humanitarian Technology Conference (GHTC), str. 1.-7., San Jose, 2017.
- [37] J. K. Patil, R. Kumar, Feature Extraction & Retrieval of Plant Leaf Diseases Using Image Histogram, Int J Electron Comput, br. 2., sv. 5, str. 143.-151., prosinac 2013.
- [38] D. Raghunath, Image Histograms in OpenCV, medium.com, dostupno na: <https://medium.com/@rndayala/image-histograms-in-opencv-40ee5969a3b7> [7.3.2020.]
- [39] M. R. Tejoindhi, B. R. Nanjesh, JagadeeshGujanuru Math, AshwinGeetD'sa, Plant Disease Analysis Using Histogram Matching Based On Bhattacharya's Distance Calculation, International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT), str. 1546.-1549., Chennai, 2016.
- [40] V. B. Jadhav, T. S. Nagwanshi, Y. P. Patil, D. R. Patil, International Research Journal of Engineering and Technology, sv. 3., str. 2076.-2078., svibanj 2016.

- [41] A. H. B. A. Wahab, R. Zahari, T. H. Lim, Detecting diseases in Chilli Plants Using K-Means Segmented Support Vector Machine, 2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC), str. 57.-61., Singapur, 2019.
- [42] S. Zhang, H. Wang, W. Huang, Z. You, Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG, Optik, sv. 157., str. 866.-872., ožujak 2018.
- [43] B. N. Manu, Plant Leaf Disease Detection and Classification using Multiclass SVM Classifier, MATLAB Central File Exchange, dostupno na: <https://www.mathworks.com/matlabcentral/fileexchange/55098-plant-leaf-disease-detection-and-classification-using-multiclass-svm-classifier> [11.5.2020.]
- [44] M. Manojkrishna, M. Neelima, M. Harshali, M. Venu Gopala Rao, Image classification using Deep learning, International Journal of Engineering & Technology, br. 2.7, sv. 7., str. 614.-617., ožujak 2018.
- [45] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press Cambridge, Cambridge, 2016.
- [46] T. Hope, Y. S. Resheff, I. Lieder, Learning tensorflow: A guide to building deep learning systems, O'Reilly Media, Inc., Sebastopol, 2017.
- [47] A Beginner's Guide to Neural Networks and Deep Learning, pathmind.com, dostupno na <https://pathmind.com/wiki/neural-network#define> [24.5.2020.]
- [48] What is a Neural Network?, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/neural-network> [24.5.2020.]
- [49] Z. Xinyi, L. Chen, Capsule graph neural network, ICLR 2019 Conference Blind Submission, New Orleans, 2019.
- [50] L. Miralles-Pechuán, D. Rosso, F. Jiménez, J. M. García, A methodology based on Deep Learning for advert value calculation in CPM, CPC and CPA networks, Soft Computing, br. 3., sv. 21., str. 651.-665., veljača 2017.

- [51] I. A. Samborska, V. Alexandrov, L. Sieczko, B. Kornatowska, V. Goltsev, M. D. Cetner, H. M. Kalaji, Artificial neural networks and their application in biological and agricultural research, J. NanoPhotoBioSciences, sv. 2., str. 14.-30., kolovoz 2014.
- [52] A. Gulli, A. Kapoor, TensorFlow 1. x Deep Learning Cookbook: Over 90 unique recipes to solve artificial-intelligence driven problems with Python, Packt Publishing Ltd, Birmingham, 2017.
- [53] Z. Meng, Y. Hu, C. Ancey, Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows, Water, br. 2., sv. 12., str. 600., veljača 2020.
- [54] W. S. Jeon, S. Y. Rhee, Plant leaf recognition using a convolution neural network, International Journal of Fuzzy Logic and Intelligent Systems, br. 1., sv. 17., str. 26.-34., ožujak 2017.
- [55] M. Loukadakis, J. Cano, M. O'Boyle, Accelerating deep neural networks on low power heterogeneous architectures, 2018.
- [56] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, H. Mahmudul, B. C. Van Essen, A. AS Awwal, V. K. Asari, A state-of-the-art survey on deep learning theory and architectures, Electronics, br. 3., sv. 8., str. 292., ožujak 2019.
- [57] B. B. Traore, B. Kamsu-Foguem, F. Tangara, Deep convolution neural network for image recognition, Ecological Informatics, sv. 48, str. 257.-268., studeni 2018.
- [58] Backpropagation, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/backpropagation> [7.7.2020.]
- [59] Konvolucija, Struna – hrvatsko strukovno nazivlje, struna.ihjj.hr, dostupno na <http://struna.ihjj.hr/naziv/konvolucija/19228/> [18.5.2020.]
- [60] D. Han, G. C. Sobabe, C. Zhang, X. Bai, Z. Wang, S. Liu, B. Guo, Spectrum sensing for cognitive radio based on convolution neural network, 2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI), str. 1.-6., Šangaj, 2017.

- [61] A. Hazarika, A. Jain, S. Poddar, H. Rahaman, Shift and Accumulate Convolution Processing Unit, TENCON 2019-2019 IEEE Region 10 Conference (TENCON), str. 914.-919., Kochi, 2019.
- [62] C. Baskin, N. Liss, E. Zheltonozhskii, A. M. Bronstein, A. Mendelson, Streaming architecture for large-scale quantized neural networks on an FPGA-based dataflow platform, 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), str. 162.-169., Vancouver, 2018.
- [63] M. Wang, B. Liu, H. Foroosh, Factorized convolutional neural networks, Proceedings of the IEEE International Conference on Computer Vision Workshops str. 545.-553., Venecija, 2017.
- [64] R. Garcia-Dias, A. Mechelli, W. H. Lopez Pinaya, S. Vieira, Autoencoders, Machine Learning: Methods and Applications to Brain Disorders, Academic Press, Cambridge, 2019.
- [65] Convolutional Neural Networks (CNN): Step 1(b) – ReLU Layer, SuperDataScience, superdatascience.org, dostupno na <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/> [10.7.2020.]
- [66] S. B. Jadhav, V. R. Udupi, S. B. Patil, Identification of plant diseases using convolutional neural networks, International Journal of Information Technology, str. 1.-10., siječanj 2020.
- [67] D. Misra, Mish: A self regularized non-monotonic neural activation function, arXiv preprint arXiv:1908.08681, kolovoz 2019.
- [68] P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, arXiv preprint arXiv:1710.05941, listopad 2017.
- [69] 7 Types of Neural Network Activation Functions: How to Choose?, missinglink.ai, dostupno na <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/#:~:text=Role%20of%20the%20Activation%20Function,neurons%20in%20the%20input%20layer.&text=The%20activation%20function%20is%20a,going%20to%20the%20next%20layer> . [4.7.2020.]
- [70] A. Gu, C. Gulcehre, T. L. Paine, M. Hoffman, R. Pascanu, Improving the Gating Mechanism of Recurrent Neural Networks, arXiv preprint arXiv:1910.09890, listopad 2019.

- [71] C. Francois, Deep learning with Python, Manning Publications Company, Shelter Island, 2017.
- [72] P. Ahamed, S. Kundu, T. Khan, V. Bhateja, R. Sarkar, A. F. Mollah, Handwritten Arabic numerals recognition using convolutional neural network, JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING, ožujak 2020.
- [73] T. Wood, Softmax Function, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer> [10.7.2020.]
- [74] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, Dive into deep learning, Unpublished Draft. Retrieved, sv. 19., str. 2019., 2019.
- [75] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, A. Rabinovich, Going deeper with convolutions. arXiv 2014, arXiv preprint arXiv:1409.4842, sv. 1409., 2014.
- [76] GoogleNet, leonardoaraujosantos.gitbooks.io, dostupno na <https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/googlenet.html> [21.6.2020.]
- [77] Inception Module, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/inception-module#:~:text=Inception%20Modules%20are%20used%20in,as%20overfitting%2C%20among%20other%20issues.> [21.6.2020.]
- [78] Vanishing Gradient Problem, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/vanishing-gradient-problem> [21.6.2020.]
- [79] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, D. Stefanovic, Solving current limitations of deep learning based approaches for plant disease detection, Symmetry, br. 7., sv. 11., str. 939., srpanj 2019.
- [80] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, Computational intelligence and neuroscience, sv. 2016., lipanj 2016.

- [81] Confusion Matrix, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/confusion-matrix> [10.7.2020.]
- [82] S. Rothe, S. F. Wirtz, D. Söffker, About the reliability of diagnostic statements: fundamentals about detection rates, false alarms, and technical requirements, Universitätsbibliothek der RWTH Aachen, Aachen, 2016.
- [83] Precision and Recall, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall> [10.7.2020.]
- [84] F-Score, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/f-score> [10.7.2020.]
- [85] What are Evaluation Metrics?, DeepAI, deepai.org, dostupno na <https://deepai.org/machine-learning-glossary-and-terms/evaluation-metrics> [10.7.2020.]
- [86] S. Kaur, S. Pandey, S. Goel, Plants disease identification and classification through leaf images: A survey, Archives of Computational Methods in Engineering, br. 2., sv. 26., str. 507.-530., travanj 2019.
- [87] V. Bhateja, S. C. Satapathy, Y. D. Zhang, V. M. Aradhya, Intelligent Computing and Communication, Springer, Bangalore, 2019.
- [88] T. Li, Y. Dou, J. Jiang, Y. Wang, Q. Lv, Optimized deep belief networks on CUDA GPUs, 2015 International Joint Conference on Neural Networks (IJCNN), str. 1.-8., Killarney, 2015.
- [89] S. Singh, A. Paul, M. Arun, Parallelization of digit recognition system using deep convolutional neural network on CUDA, 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), str. 379.-383., Chennai, 2017.
- [90] A. Parvat, J. Chavan, S. Kadam, S. Dev, V. Pathak, A survey of deep-learning frameworks, 2017 International Conference on Inventive Systems and Control (ICISC), str. 1.-7., Coimbatore, 2017.

- [91] S. Mayer, H. V. Le, N. Henze, Machine learning for intelligent mobile user interfaces using TensorFlow, Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, str. 1.-5., New York, 2017.
- [92] W. G. Hatcher, W. Yu, A survey of deep learning: platforms, applications and emerging research trends, IEEE Access, sv. 6., str. 24411.-24432., 2018.
- [93] A. Luashchuk, Why I Think Python is Perfect for Machine Learning and Artificial Intelligence, medium.com, dostupno na <https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6> [19.5.2020.]
- [94] Anaconda, anaconda.com, dostupno na <https://www.anaconda.com/> [19.5.2020.]
- [95] MATLAB Overview, MathWorks, mathworks.com, dostupno na <https://www.mathworks.com/products/matlab.html> [19.5.2020.]
- [96] TensorFlow, TensorFlow, tensorflow.org, dostupno na <https://www.tensorflow.org/about> [20.5.2020.]
- [97] D. Padua, Data Flow Graphs, SpringerLink – Encyclopedia of Parallel Computing, dostupno na https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-09766-4_294 [20.5.2020.]
- [98] Tenzor, Hrvatska enciklopedija, enciklopedija.hr, dostupno na <https://www.enciklopedija.hr/natuknica.aspx?ID=60828> [20.5.2020.]
- [99] Keras, keras.io, dostupno na <https://keras.io/> [20.5.2020.]
- [100] NumPy, Numpy.org, dostupno na <https://numpy.org/> [20.5.2020.]
- [101] F. Malik, Why should we use NumPy?, medium.com, dostupno na <https://medium.com/fintechexplained/why-should-we-use-numpy-c14a4fb03ee9> [20.5.2020.]
- [102] Scientific Applications, Python Guide, dostupno na <https://docs.python-guide.org/scenarios/scientific/> [20.5.2020.]
- [103] OpenCV, opencv.org, dostupno na <https://opencv.org/about/> [20.5.2020.]

[104] J. VanderPlas, Visualization with Matplotlib, github.io, dostupno na <https://jakevdp.github.io/PythonDataScienceHandbook/04.00-introduction-to-matplotlib.html> [20.5.2020.]

SAŽETAK

U teorijskom dijelu ovoga rada opisan je problem raspoznavanja bolesti lista biljke uz prikaz mogućnosti primjene postupaka dubokog strojnog učenja. Razvoj modela za klasifikaciju bolesti potaknuo je unaprjeđenje automatiziranih sustava za otkrivanje biljnih bolesti pomoću vidljivih simptoma na lišću dok je sam proces izdvajanja značajki ugrađen u algoritam. Znanje o ovom procesu značajno bi doprinijelo području znanosti pri razumijevanju dijagnostičkih postupaka i detekciji bolesti. U eksperimentalnom dijelu rada proveden je postupak klasifikacije bolesti lista biljke na prerađenom GoLeNet i VGGNet modelu te MobileNet, InceptionV3 i InceptionResNetV2 mrežnim arhitekturama. Sve konvolucijske neuronske mreže trenirane su i testirane na odabranom skupu podataka koji uključuje bolesne i zdrave jedinice kukuruza podijeljene u četiri klase. Najpreciznije rezultate omogućio je InceptionResNetV2 model s 98.9% točnosti na odabranom skupu podataka.

Ključne riječi: dijagnosticiranje bolesti biljke, duboko strojno učenje, klasifikacija, konvolucijske neuronske mreže, prijenosno učenje

ABSTRACT

Plant leaf disease recognition using image-based machine learning

The problem of recognizing plant leaf diseases with a review of the current possibilities by applying deep machine learning is described in the theoretical part of this thesis. The development of disease classification models has prompted improvements of automated plant disease detection systems using visible symptoms, while the feature extraction is built in the algorithms. Knowledge of this process has contributed to the field of science in understanding diagnostic procedures and disease detection. In the experimental part of the thesis, the procedure of leaf diseases classification with the modified GoogleNet and VGGNet model MobileNet, alongside InceptionV3 and InceptionResNetV2 network architectures was performed. All convolutional neural networks were trained and tested on selected set data that included diseased and healthy maize images divided into four classes. The most accurate results were achieved by the InceptionResNetV2 model with 98.9% accuracy on the selected data set.

Keywords: diagnosing plant diseases, deep machine learning, classification, convolutional neural networks, transfer learning

ŽIVOTOPIS

Petra Radočaj rođena je 20.6.1998. godine u Našicama, s prebivalištem u Đakovu. Nakon završene prirodoslovno - matematičke gimnazije u Đakovu upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na Preddiplomski sveučilišni studij Računarstvo 2017. godine na koji ostvaruje izravan upis zbog izvrsnog uspjeha.

Dobitnica je stipendija za darovite učenike, sveučilišne stipendije za izvrsnost te STEM stipendija. Demonstratorica na kolegijima Programiranje 1 i Programiranje 2. Trenutno zaposlena kao Software Developer u tvrtki „NTH Group“ u odjelu *Payment & Messaging*. Sudjelovala na „OSC Hackathon 2015.“ u kategoriji „Juniora“ uz volontiranje u Code Clubu Gimnazije. Zainteresirana je za duboko strojno učenje te primjenu računarstva u medicini za koje polaže certifikate.

PRILOZI

1. „Raspoznavanje bolesti lista biljke uporabom strojnog učenja na temelju slike“ u .docx formatu
2. „Raspoznavanje bolesti lista biljke uporabom strojnog učenja na temelju slike“ u .pdf formatu
3. Izvorni kod