Prepoznavanje bolesti vinove loze sa slika

Markota, Krešimir

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

Sveučilišni studij

PREPOZNAVANJE BOLESTI VINOVE LOZE SA SLIKA

Diplomski rad

Krešimir Markota

Osijek, 2020. godina

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 28.09.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Krešimir Markota
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1001R, 24.09.2019.
OIB studenta:	04392871853
Mentor:	Izv. prof. dr. sc. Emmanuel Karlo Nyarko
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. . Damir Filko
Član Povjerenstva 1:	Izv. prof. dr. sc. Emmanuel-Karlo Nyarko
Član Povjerenstva 2:	Dr.sc. Petra Đurović
Naslov diplomskog rada:	Prepoznavanje bolesti vinove loze sa slika
Znanstvena grana rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Treba istražiti prikladnu metodu za detekciju i prepoznavanje bolesti vinove loze sa slika. Implementirati predloženi postupak kao softversko rješenje te testirati na odgovarajuću bazu slika. Tema rezervirana za: Krešimir Markota
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	28.09.2020.

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, 06.10.2020.

Ime i prezime studenta:	Krešimir Markota
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1001R, 24.09.2019.
Turnitin podudaranje [%]:	13

Ovom izjavom izjavljujem da je rad pod nazivom**: Prepoznavanje bolesti vinove loze sa slika**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora

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

Vinova loza je jedna od najrasprostranjenije uzgajanih poljoprivrednih kultura širom svijeta. Unatoč blagodatima koje pruža vinova loza, zahtjevna je za uzgajanje te ju je često potrebno braniti od raznih nametnika i bolesti koje ju napadaju. Neke od najpoznatijih bolesti vinove loze su: plamenjača, crna trulež i apopleksija vinove loze. Vinari koji se profesionalno bave ovom kulturom ulažu veliki dio novčanih sredstava u obranu od bolesti. Stoga, rano otkrivanje bolesti vinove loze može potencijalno smanjiti trošak i poboljšati kvalitetu ploda. Upravo zbog toga, metode automatskog prepoznavanja bolesti vinove loze su prijeko potrebne.

Ubrzanim razvojem računalne tehnologije tradicionalne metode strojnog učenja su sve više primjenjive u predviđanju bolesti biljke. Iako su se tradicionalne metode strojnog učenja pokazale vrlo korisnima u prepoznavanju i dijagnostici bolesti usjeva, one su ograničene na primjenu sekvencijalnih postupaka kao što su: segmentacija slike, izvlačenje svojstava i prepoznavanje značajki. Metode dubokog učenja imaju odličnu sposobnost poopćavanja i robusnosti u drugim područjima kao što su: obrada signala, prepoznavanje lica, prepoznavanje rupa na cesti, analiza fotografije u biomedicini.

Drugo poglavlje prikazuje već postojeća slična dana rješenja na temu ovog diplomskog rada te se spominju poznati modeli neuronskih mreža. Treće poglavlje daje informacije o kategorijama bolesti vinove loze koje će se proučavati, te se također spominju korišteni alati i tehnologije za izradu sustava za prepoznavanje bolesti vinove loze. U četvrtom poglavlju se opisuje predobrada podatkovnog skupa i arhitektura konvolucijskih neuronskih mreža. Posljednje, peto poglavlje daje na uvid rezultate postojećih rješenja.

1.1. Zadatak diplomskog rada

Zadatak diplomskog rada je istražiti prikladnu metodu za detekciju i prepoznavanje bolesti vinove loze sa slika. Nadalje, treba implementi predloženi postupak kao rješenje programske podrške te testirati na odgovarajućoj bazi podataka slika. Aplikacija treba na temelju zadanog podatkovnog skupa slika, klasificirati bolesti.

2. Pregled metoda prepoznavanja bolesti biljaka

Kako bi se smanjile štete od bolesti, mnogi istraživači su uložili velike napore pri prepoznavanju raznih bolesti biljaka. Stalnim razvojem algoritama strojnog učenja počela je njihova široka primjena pri prepoznavanju biljnih štetnika i bolesti. Tehnike strojnog učenja primijenjene su za klasifikaciju bolesti biljaka već u počecima svog nastajanja. Prema [7], predložena metoda za prepoznavanje i segmentiranje bolesti peronospore i pepelnice temeljila se na K-means klasteriranju pri segmentaciji bolesti i SVM klasifikatoru za prepoznavanje navedenih bolesti sa stopama prepoznavanja od 90% za peronosporu i 93,33% za pepelnicu. Značajan napredak postignut je u korištenju pristupa obradi slike za otkrivanje različitih bolesti usjeva. Prema [8] primijenjena je neuronska mreža kako bi se sliku lista krumpira kategoriziralo kao zdravu ili bolesnu. Njihovi rezultati pokazali su da neuronska mreža s unazadnom propagacijom može učinkovito otkriti mjesta bolesti te klasificirati određenu vrstu bolesti s točnošću od 92%. Razvoj dubokih konvolucijskih neuronskih mreža doveo je do proboja u klasifikaciji biljnih bolesti, koje mogu pronaći vrlo veliku varijaciju patoloških simptoma u vizualnom pogledu, čak i veliku različitost unutar klase i malu sličnost među klasama koju samo možda primjećuju botaničari. Nadalje, prema [10], predložen je pristup konvolucijskim neuronskim mrežama pri prepoznavanju slika lišća biljaka gdje je prosječna točnost iznosila 99,7% na podatkovnom skupu koji je sadržavao 44 različite vrste, međutim skalabilnost podatkovnog skupa bila je veoma mala. Također postoji slučaj [11] gdje su modeli dubokog učenja fino podešeni prilikom njihovog ponovnog treniranja nad poznatim *ImageNet* skupom podataka kako bi se prepoznalo 14 različitih usjeva i 26 različitih bolesti. Modeli su bili procijenjeni na javno dostupnom skupu podataka koji uključuje 54,306 slika bolesnih i zdravih listova biljaka koji su prikupljeni pod kontroliranim uvjetima. Postignuta najbolja točnost bila je 99,35%.

2.1. Poznati modeli neuronskih mreža

Iako su osnovni okviri konvolucijskih neuronskih mreža, kao što su : *AlexNet* [12], *VGGNet* [13], *GoogLeNet* [14], *DenseNet* [15] i *ResNet* [16] pokazali učinkovitost i postigli široku primjenu prepoznavanju bolesti usjeva, većina prethodnih radova ima otežanje pri postizanju većih mjera točnosti klasifikacije. Zapravo, jedan model ne može više ispuniti daljnje zahtjeve u smislu preciznosti. Na velikim natjecanjima u strojnom učenju najbolji se rezultati obično postižu međusobnim spajanjem više modela, a ne jednim modelom. Na primjer, dobro poznati *Inception-ResNet-v2* [17] nastao je spajanjem dvije odlične duboke konvolucijske neuronske mreže kao što joj samo ime govori.

3. TEORIJSKA PODLOGA I PRIMIJENJENE TEHNOLOGIJE

U ovom poglavlju su iznesene ideje rada, način rada neuronskih mreža i također primijenjene biblioteke za dizajniranje i treniranje istih, a isto tako primijenjeni alati koji su služili pri izradi ovog diplomskog rada.

3.1. Idejno rješenje

Ideja ovog rada je razviti sustav za prepoznavanje bolesti vinove loze pomoću podatkovnog seta fotografija listova vinove loze koji su podijeljeni u četiri kategorije.

U navedene četiri kategorije se svrstavaju:

1.crna trulež vinove loze,

2. apopleksija vinove loze (*Esca*),

3. bljedilo lista (*Leaf blight*),

4. primjeri uzoraka zdravih listova.

Sustav bi, na danu novu fotografiju vinove loze koja nije u skupu treniranih fotografija, trebao prepoznati o kojoj se bolesti radi iz zadanog skupa navedenih kategorija. Ako je primjerak dane fotografije zdravi list, sustav bi trebao prepoznati da se radi o zdravom uzorku. Potrebno je uraditi predobradu fotografija kako bi fotografije bile proslijeđene u pogodnom formatu za treniranje neuronskih mreža. Cijeli sustav je temeljen na korištenju *Python* biblioteka otvorenog koda.

3.2. Neuronske mreže

Strojno učenje je područje umjetne inteligencije koje primjenjuje statistiku kako bi se pronašle značajke u velikoj količini podataka bez dodatne potrebe za programiranjem određenih zadataka. Ključna ideja strojnog učenja je pisanje algoritama koji uče na temelju podataka i rade predviđanja odnosno estimaciju nad podacima. Postoje tri različite kategorije strojnog učenja.

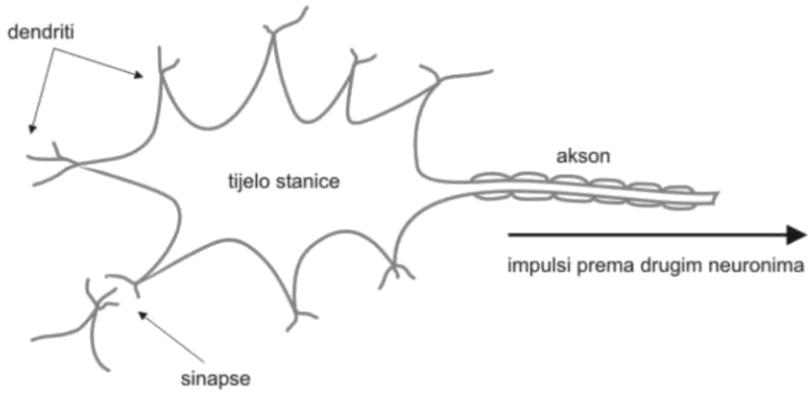
1. U nenadziranom učenju stroj uči na temelju ulaznih podataka i sam pronalazi smisleni sastav podataka bez vanjskog nadzora.
2. U nadziranom učenju stroj uči na temelju ulaznih podataka kojima su pridruženi pripadajući izlazni podaci s ciljem izvršavanja smislene predikcije nad novim podacima koji se ne nalaze u već poznatom skupu podataka.

3. U podržanom učenju stroj djeluje kao agent koji ima interakciju s okolinom, traži ponašanja koja donose nagradu i na temelju iskustva postaje uspješniji.

Neuronske mreže se često povezuju s pojmovima umjetne inteligencije, strojnog učenja i dubokog učenja. Umjetna inteligencija područje je vrlo velike domene gdje se strojevima nastoji usaditi kognitivne sposobnosti kao što su: zaključivanje i dedukcija, učenje, rješavanje problema, percepcija, računalni vid i mnoge druge. Jednostavno rečeno, umjetna inteligencija predstavlja bilo koju radnju gdje strojevi nastoje oponašati inteligentno ponašanje koje ljudi obično pokazuju. Umjetna inteligencija sadrži elemente informatike, matematike i statistike. Neka važna svojstva neuronskih mreža naspram konvencionalnih načina obrade podataka su sljedeće:

- Rade s velikim brojem parametara i varijabli
- Prilagođavaju se okolini
- Formiraju znanje učeći iz velikog broja primjera
- Vrlo dobro procjenjuju.

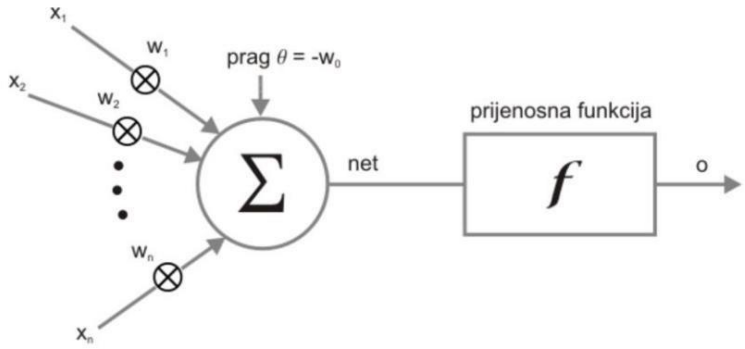
Neuronske mreže su niz algoritama koji nastoje prepoznati temeljne odnose u skupu podataka kroz proces koji oponaša način rada ljudskog mozga. U tom se smislu neuronske mreže odnose na sustave neurona, bilo organskih ili umjetnih. One se mogu prilagoditi promjenama ulaznih podataka kako bi ista ta mreža generirala najbolji mogući rezultat bez potrebe za promjenama u dizajnu izlaznih kriterija. Prve neuronske mreže pojavljuju se 1943. godine kada su Warren McCulloch i Walter Pitts opisali svoje ideje i napravili za model jednostavnu neuronsku mrežu koristeći električne krugove [2]. Izvorna ideja bila je stvaranje računalnog sustava koji bi rješavao probleme na način koji ljudski mozak rješavao probleme. Međutim, vremenom su se istraživači usmjerili na rješavanje specifičnih zadataka, što je dovelo do odstupanja od biološkog pristupa. Tako da se danas neuronske mreže koriste u područjima marketinga, robotike, medicine i mnogim drugim.



Slika 3.1.: Struktura neurona.[1]

Slika 3.1 prikazuje izgled ljudskog neurona koji se sastoji od:

- Sinapsi - spojno mjesto komunikacije između dva neurona
- Dendrita - primatelji informacije od drugih neurona utječu na potencijal stanice
- Tijela stanice – Spremnik informacije u obliku električnog potencijala između vanjskog i unutrašnjeg dijela stanice
- Aksona – izdužena cjevčica, prenosi električne impulse.



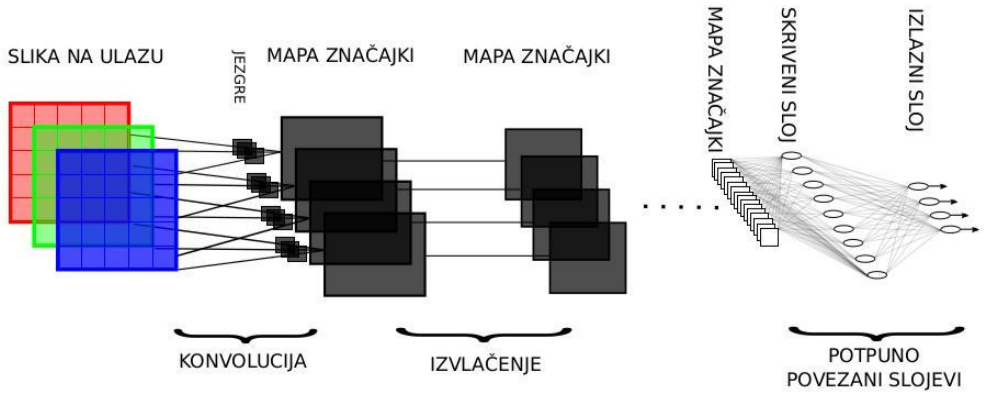
Slika 3.2.: Struktura aktivacijske funkcije[1]

Slika 3.2. prikazuje strukturu umjetnog neurona koji se još naziva čvor ili jedinica u cijeloj neuronskoj mreži. Ovakva struktura pokušava oponašati rad ljudske moždane stanice. Kod ovakve strukture signali su brojčane vrijednosti, jakost prijenosa signala sinapse se opisuje težinskom vrijednošću ili faktorima *w,* tijelo stanice predstavlja zbrajalo ulaznih vrijednosti (*x1, x2, x3 , ... xn)* koje su pomnožene težinama (*w1,w1,w3, ... wn)* a na izlazu daje vrijednost *net.Net predstavlja ulaz u aktivacijsku funkciju.* Ako je zbroj otežanih vrijednosti signala veći od očekivanog praga osjetljivosti neurona, aktivacijska funkcija postavlja izlaz neurona na vrijednost 1, a ako zbroj otežanih vrijednosti

signala manji od očekivanog praga osjetljivosti neurona, tada aktivacijska funkcija postavlja izlaz neurona na vrijednost 0.

3.2.1. Konvolucijske neuronske mreže

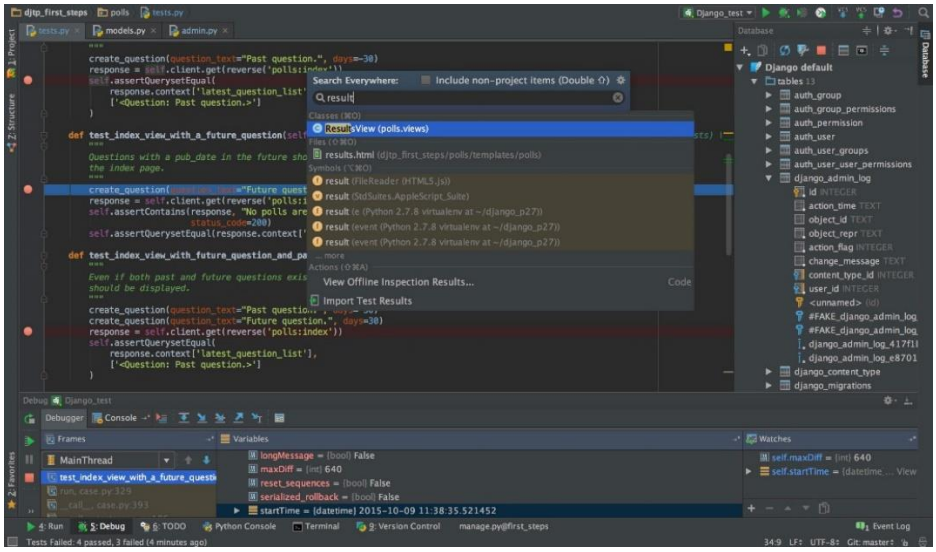
Konvolucijska neuronska mreža je tip neuronskih mreža koji prvenstveno služi za klasifikaciju i prepoznavanje objekata sa slika. Konvolucijska neuronska mreža uzima sliku na ulazu. Slika može biti višekanalna u boji ili monokromatska. Struktura konvolucijskih neuronskih mreža je obično takva da nakon ulaza naizmjence slijede konvolucijski slojevi i slojevi sažimanja. U nastavku na slojeve konvolucije i sažimanja nastavlja se nekoliko potpuno povezanih slojeva koji predstavlja klasični perceptron. Potpuno povezani slojevi su jednodimenzionalni te uključuju i zadnji izlazni sloj koji sadrži onoliki broj neurona koliko ima klasa za prepoznavanje. Prema [21], tipične arhitekture konvolucijskih neuronskih mreža sadrže oko desetak slojeva. Slojevi konvolucije i sažimanja su dvodimenzionalni i takvi oblici neurona nazivaju se mapama značajki (engl. *feature maps*). Mape značajki u svakom sloju postaju manjih dimenzija. Zadnji sloj, iako je i dalje formalno dvodimenzionalan, dimenzija je 1 x 1 te je poveznica na perceptron u zadnjim slojevima konvolucijske neuronske mreže. Slojevi konvolucije imaju uvijek jedan-na-više vezu sa slojem ispred te uči težine koje su spremljene u jezgrama (engl. *kernels*) s kojima obavljaju također konvoluciju. Slojevi sažimanja imaju uvijek jedan-na-jedan vezu s prethodnim slojem te ne uče nikakve vrijednosti. Prema [29], zbog brzine i jednostavnosti konvolucijskih neuronskih mreža, jedan od najpopularnijih odabira je višeslojna unaprijedna neuronska mreža sa sigmoidalnim aktivacijskim funkcijama u konačnim slojevima konvolucijskih neuronskih mreža. Slika 3.3 prikazuje opću strukturu opisanih konvolucijskih neuronskih mreža.



Slika 3.3.: Prikaz opće strukture konvolucijskih neuronskih mreža[21]

3.3. Pycharm

Pycharm je integrirano razvojno okruženje koje se primjenjuje u računalnom programiranju, posebno za jezik Python. Razvila ga je tvrtka iz Češke pod nazivom *JetBrains.* Omogućuje analizu koda, grafički program za uklanjanje koda, integraciju sa sustavima za upravljanje verzijama koda, a isto tako podržava razvoj mrežnih aplikacija s Django programskim okvirom i povezivanje s Anacondom koja se primjenjuje u podatkovnim znanostima. Pycharm podržava razvoj aplikacija na više platformi i dolazi u dvije različite inačice. Prva inačica je profesionalna inačica koja sadrži dodatke za razvoj znanstveni i mrežnih aplikacija s podrškom za HTML, JavaScript i SQL. Druga inačica predstavlja inačicu šire Python zajednice koja je besplatna i podržava čisto Python programiranje[4].



Slika 3.4.: Prikaz PyCharm okruženja.[4]

3.4. Python

Python je programski jezik opće namjene i visoke razine. Python ima automatsku memorijsku alokaciju i podržava dinamično pridruživanje tipova podataka. Sintaksa ovog programskog jezika je lako čitljiva, što ga čini pogodnim za usavršavanjem programiranja među početnicima. Međutim, unutar programerske zajednice česte su kritike na račun njegove sporosti u izvođenju koda. S obzirom na to da je Python interpreterski jezik, programi napisani u njemu se vrše sporije nego u ostalim jezicima koji rade na principu kompajlera. Python podržava više programskih paradigmi uključujući strukturno, objektno orijentirano i funkcijsko programiranje[6].

3.5. Tensorflow

Tensorflow je biblioteka otvorenog koda za numeričko računanje i strojno učenje širokog raspona. Izvorno ga je razvio *Google Brain* tim unutar Google-ove organizacije za istraživanje strojnog učenja i dubokih neuronskih mreža, ali sustav je dovoljno općenit da se može primjeniti i u drugim znanstvenim područjima. *Tensorflow* se može pokretati na više različitih platformi, a radi gotovo na svim vrstama sklopovlja. Nastao je kao biblioteka otvorenog koda kako bi se velikom dijelu zajednice omogučio sudjelovanje u razvijanju *Tensorflow* biblioteke. Verzija zadnje inačice je *Tensorflow 2.0* [22].

3.6. Keras

Keras je *API* (engl. *Application Programming Interface*) za duboko učenje, napisan u Python programskom jeziku. Pokreće se na vrhu TensorFlow platforme i razvijen je s naglaskom na omogućavanje brzog eksperimentiranja, tj. na sposobnost što bržeg prijelaza od ideje do rezultata. Keras je u početku razvijen kao dio projekta ONEIROS (*Open-end Neuro-Electronic Intelligent Robot Operating System*) s ciljem brzog eksperimentiranja s neuronskim mrežama. Prema [3], glavne značajke ove biblioteke zbog kojeg je odabrana pri izradi u ovom radu su:

- Modularnost - modeli neuronskih mreža grade se kao slijed grafova koji se mogu kombinirati na lak način
- Minimalizam – spomenuta biblioteka napisana je u Pythonu i svaki modul je kratak i sam po sebi intuitivan za korištenje
- Laka proširivost – veoma jednostavno dopunjavanje novih funkcija i modula.

3.7. OpenCV

OpenCV(*Open Source Computer Vision Library*) je biblioteka napisana u C++ i Python programskim jezicima. Otvorenog je koda i služi za upotrebu pri rješavanju problema strojnog učenja i računalnog vida. Navedena biblioteka osmišljena je kako bi ubrzala primjenu općenite infrastrukture za računalni vid u poslovne svrhe. Biblioteka sadrži više od 2500 optimiziranih algoritama, što obuhvaća široki raspon klasičnih i najsuvremenijih algoritama računalnog vida i strojnog učenja. Ti se algoritmi primijenjuju za prepoznavanje lica i raznih objekata, klasificiranje ljudskih radnji u videozapisima, praćenje objekata u pokretu, izvlačenje značajki sa slika i drugo [7].

3.8. Matplotlib

Matplotlib je biblioteka za izradu 2D crteža u Pythonu. Iako svoje podrijetlo vuče od grafičkih naredbi MATLAB programskog jezika, neovisan je o njemu i može se primijeniti na Python načine u obliku objektno orijentirane paradigme. Iako je Matplotlib biblioteka napisana prvenstveno u čistom Python programskom jeziku, intenzivno primjenjuje NumPy i ostale module kako bi pružio dobre izvedbe čak i za velike nizove programskih struktura. Matplotlib je dizajniran s namjerom da bi se trebali stvoriti jednostavni crteži sa samo nekoliko naredbi.

3.9. Ostale Python biblioteke

Osim navedenih Python modula, ostali važniji primijenjeni Python moduli su:

- NumPy - glavni paket za znanstveno računanje u Pythonu [25]
- Random -implementira pseudo slučajne brojeve za razne varijacije matematičkih distribucija [26]
- Pickle - omogućuje pretvaranje Python objekata u tok bajtova [27]
- Time - omogućuje predočavanje vremena u programskom kodu na više načina [28].

4.REALIZACIJA RJEŠENJA

Prepoznavanje bolesti vinove loze dugi niz godina se vrši od strane čovjeka, takav postupak je subjektivan, sklon greškama, vremenski zahtjevan i skup proces. Iako se čini da je takav posao djelomično lak za čovjeka zbog njegove urođene sposobnosti vida, za računalo je ono vrlo zahtjevan i složen proces. Stoga se u ovom poglavlju daje na uvid realizacija rješenja. Rješenje je realizirano kroz nekoliko glavnih koraka koji podrazumijevaju pronalazak i preuzimanje skupa podataka slika bolesti vinove loze, predobradu skupa podataka, uspostavljanje arhitekture konvolucijske neuronske mreže i treniranje iste nad skupom podataka.

4.1. Preuzimanje skupa podataka slika i njegova predobrada

Za potrebe ovog rada, skup podataka slika preuzet je s *Kaggle* mrežne stranice. *Kaggle* je mrežna stranica koja sadrži velik broj skupova podataka iz različitih disciplina nad kojima se mogu primjenjivati različite statističke metode i metode strojnog učenja kako bi se mogle izvući potrebite i korisne informacije iz istih. Za potrebe ovog rada preuzet je *PlantVillageDataset* skup podataka koji sadrži 38 različitih kategorija listova biljaka s različitim vrstama bolesti na sebi. Od 38 različitih klasa odabrane su četiri kategorije za potrebe ovog rada od toga tri su kategorije bolesti vinove loze koje se pokazuju na listovima i jedna kategorija koja predstavlja zdrave listove. Svaka od spomenutih kategorija sadrži oko 1000 slika. Slika 4.1 prikazuje reprezentativne uzorke kategorija iz podatkovnog skupa a to su redom s lijeva na desno:

1.crna trulež vinove loze,

2. apopleksija vinove loze (lat. *Esca*),

3. bljedilo lista (engl. *Leaf blight*),

4. primjeri uzoraka zdravih listova.

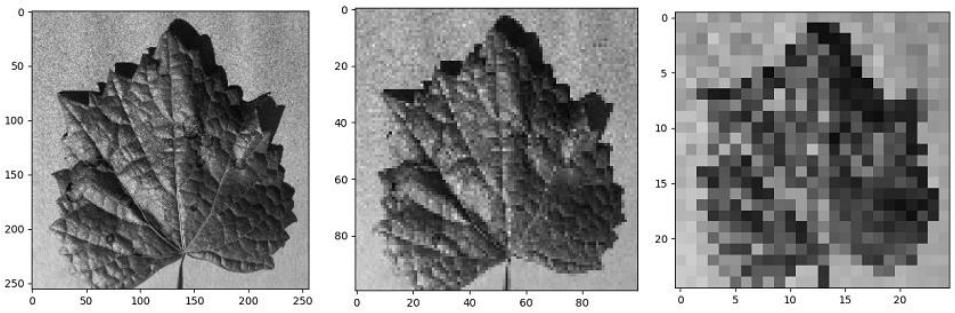
Slika 4.1.: Prikaz reprezentativnih uzoraka navedenih kategorija iz podatkovnog seta

Za potrebe testiranja rezultata iz svake kategorije izdvojeno je po 100 slika kako se izdvojene slike ne bi našle u skupu slika nad kojima će se trenirati neuronska mreža. Prije treniranja konvolucijske neuronske mreže potrebno je predobraditi slike kako bi se u pogodnom formatu za treniranje konvolucijskih neuronskih mreža proslijedile istima na ulaz. Prva dva koraka pri predobradi podataka su:

1. Pretvaranje višekanalne slike u boji u monokromatsku sliku,
2. Smanjivanje dimenzija slike.

Slika 4.2.: Prikaz primjera monokromatske slike

Prvim korakom se iz tenzora slike izbacuju tri kanala koji predstavljaju boje i dobiva se monokromatska slika, time je veličina slike smanjena tri puta što smanjuje vrijeme treniranja konvolucijske neuronske mreže. Oba ova koraka izvršavaju se kako bi se smanjilo vrijeme treniranja konvolucijske neuronske mreže. Pri drugom koraku treba paziti da se dimenzije slika ne smanje previše jer se tako mogu izgubiti karakteristične značajke slike koje su potrebne za daljnje prepoznavanje. Također sve slike koje se prosljeđuju na treniranje konvolucijske neuronske mreže moraju imati iste dimenzije kako bi ih klasifikator mogao prepoznati. Slika 4.3. prikazuje promjenu kvalitete smanjenjem dimenzija jedne te iste slike.



Slika 4.3.: Prikaz smanjenja dimenzije slike

Idući korak pri predobradi podataka je njihovo međusobno miješanje. Ovaj korak je vrlo bitan jer se njime izbjegava zbunjivanje klasifikatora pri treniranju konvolucijske neuronske mreže, pomaže pri bržem konvergiranju treninga konvolucijske neuronske mreže, sprječava pristranost modela nekoj klasi i sprječava model da nauči redoslijed treniranja i samim time izbjegne prekomjerni trening (engl. *overfitting).* Ako je procijenjeno da trenutna količina slika nije relevantna za treniranje konvolucijske neuronske mreže, trenutni podatkovni set slika može se umjetno proširiti. Postoje gotove funkcije unutar Keras biblioteke koje omogućavaju navedenu radnju. Umjetno proširenje podataka radi tako da se već postojeće slike u skupu podatka mogu modificirati tako da se promijeni njihova trenutna vrijednost ali da se i ne izgube potrebne značajke. U tu svrhu, slike se mogu, okretati oko osi za zadani kut, pomicati vertikalno i horizontalno za zadanu vrijednost, te povećavati i smanjivati. Tako se trenutni skup podataka može proširiti i za nekoliko puta. Umjetno proširenje podatkovnog skupa slika posebice je dobro raditi u trenucima kada nedostaje relevantan broj podataka ili kada je nemoguće na neki drugi način proširiti podatkovni skup. Zadnji korak predobrade podataka prije prosljeđivanja podataka na treniranje neuronskoj mreži je normalizacija podataka. Normalizacija podataka skalira podatkovne vrijednosti slika s vrijednosti između 0 i 255 na vrijednosti između 0 i 1, a ona se vrši tako da se trenutne vrijednosti slika podijele s 255. Korak normalizacije ubrzava konvergiranje treniranja modela i povećava njegovu točnost. Nakon što se odradi predobrada podataka, podatke je potrebno spremiti u zasebnu datoteku kako bi se izbjeglo ponovno izvršavanje cijelog procesa predobrade podataka.

4.2. Izgradnja arhitekture konvolucijske neuronske mreže

S obzirom na to da su neuronske mreže nepredvidive metode dobivanja optimalnih rješenja, bilo je potrebno izgraditi arhitekturu konvolucijske neuronske mreže tako da se na predobrađenim podacima istrenira više varijacija jedne konvolucijske neuronske mreže kako bi se moglo doći do zaključka koja varijacija konvolucijske neuronske mreže daje najbolje rezultate.

Klasična struktura konvolucijske neuronske mreže je postavljena a parametri koji su se izmjenjivali kroz petlje su:

1. Broj neurona u svakom sloju koji može biti 32, 64 i 128
2. Broj konvolucijskih slojeva koji može biti 1, 2, 3
3. Broj potpuno povezanih slojeva koji može biti 0, 1, 2.

Na kraju arhitekture neuronske mreže nalazi se potpuno povezani sloj koji sadrži četiri neurona koji predstavljaju četiri kategorije koje se primjenjuju za klasifikaciju u ovom radu, a *softmax* aktivacijska funkcija pretvara izlaze u vjerojatnosti. *Softmax* aktivacijska funkcija je preporučeni izbor kod problema klasifikacije zbog svojstva da suma vjerojatnosti na izlazu daje vrijednost jedan.

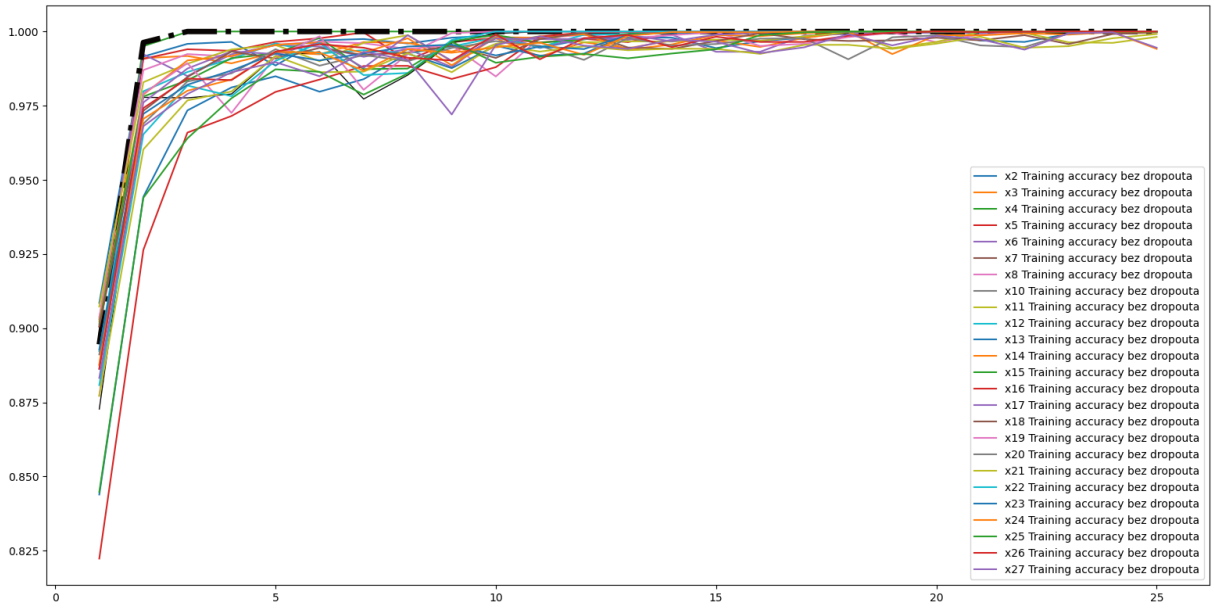
Prije početka treniranja arhitekture konvolucijske neuronske mreže, treba se namjestiti proces učenja, što se vrši funkcijom *compile().* Navedenoj metodi se kao parametri predaju:

- Algoritam optimizacije – identifikator postojećeg algoritma ili instanca klase *Optimizer*
- Funkcija gubitka – cilj ove funkcije je postići minimalnu vrijednost uz pomoć algoritma optimizacije, također može biti zadana ugrađenom funkcijom gubitka
- Popis metrika – za klasifikacijske probleme ovaj parametar postavlja se na „accuracy“, a njegova vrijednost se dobiva kao odnos točno klasificiranih podataka i ukupnog broja podataka.

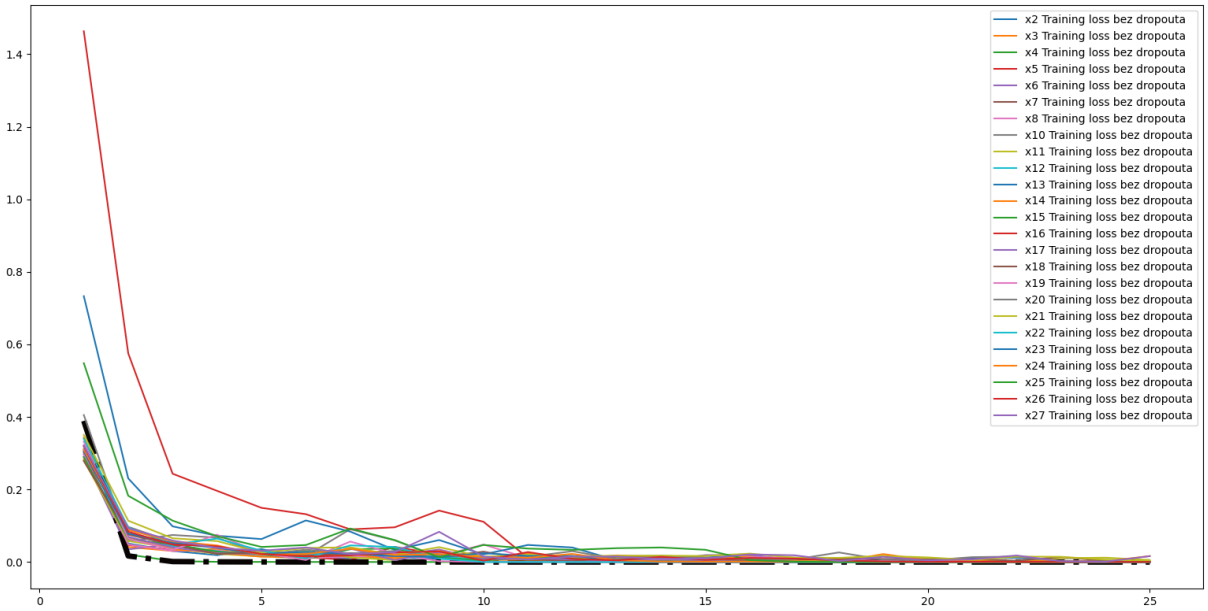
Sam model se trenira funkcijom *fit()* koja kao argumente prima ulazne podatke i klasu kojoj pripada ulazni podatak. Svi se podaci predaju na treniranje u obliku *Numpy* nizova. Kao argument metodi *fit()* također se može predati broj epoha, lista metoda povratnog poziva (engl. *callback function*), validacijski skup podataka i tako dalje. Metoda *fit()* vraća i objekt klase *History* gdje su spremljene varijable pogreške i metričke vrijednosti uspješnih epoha na trening skupovima te isto tako i na validacijskim skupovima.

5.TESTIRANJE I REZULTATI

Nakon izgradnje arhitekture konvolucijske neuronske mreže potrebno je iste istrenirati i mjeriti rezultate, te dobivene rezultate spremiti kako bi se mogli prikazati u grafičkom obliku. U obzir su uzeti rezultati pri treniranju konvolucijskih neuronskih mreža koje su na kraju ulaznog sloja, konvolucijskog sloja i potpuno povezanog sloja imali metodu *dropout(),* koja nasumično odbacuje zadani postotak neurona radi mogućeg postizanja boljih rezultata. Također su u obzir uzeti rezultati pri treniranju konvolucijskih neuronskih mreža koje na kraju ulaznog sloja, konvolucijskog sloja i potpuno povezanog sloja nisu imali izvršavanje *dropout()* metode. U oba spomenuta slučaja arhitekture su se trenirale kroz 25 epoha te je tako dobiveno 27 različitih kombinacija istreniranih modela za svaki slučaj, što je ukupno 54 različite kombinacije istreniranih modela. Rezultati su pokazali da su svi modeli nakon 25 epoha treniranja postigli zadovoljavajuću točnost od preko 98% na trening skupu. S toga su za testiranje odabrani modeli koji su postigli najbržu konvergenciju željenog rezultata. Slike 5.1 i 5.2 prikazuju konvergiranje funkcije točnosti, odnosno gubitka prema zadovoljavajućem rezultatu od preko 98% za funkciju točnosti i 2% za funkciju gubitka. Modeli sa slika 5.1 i 5.2 su trenirani na arhitekturi bez izvršavanja metode *dropout(),* a model koji je odabran kao model s najboljim konvergiranjem k željenom rezultatu podebljan je crnom bojom.

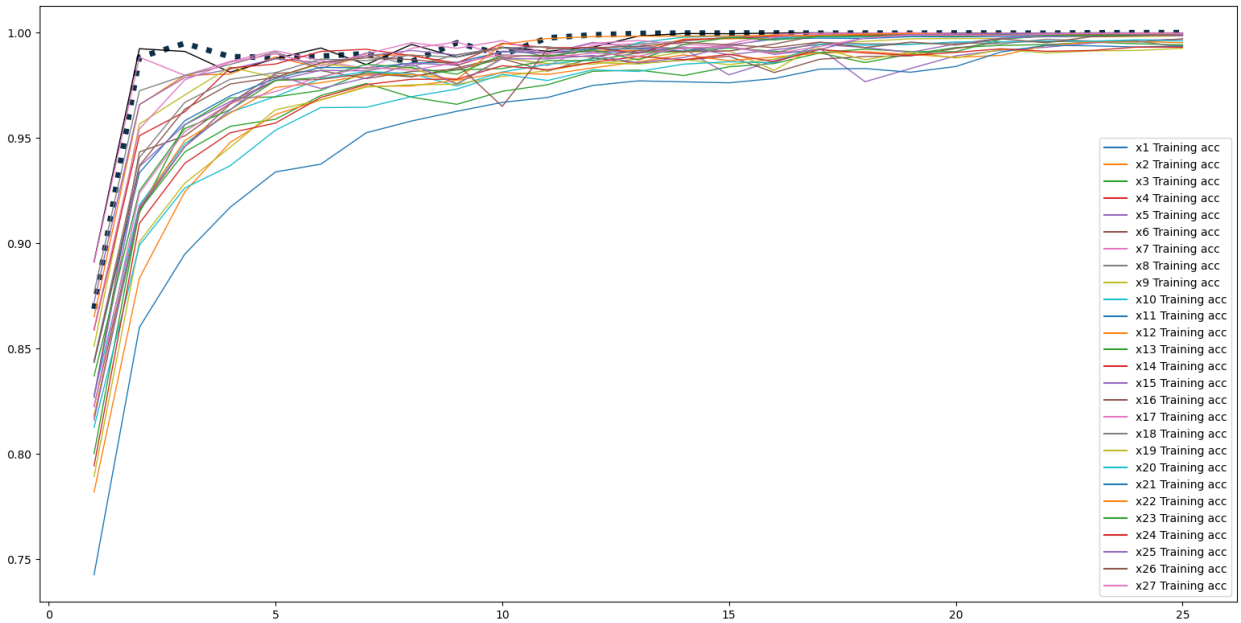


Slika 5.1.: Prikaz konvergiranja funkcije točnosti bez izvršavanja dropout() metode

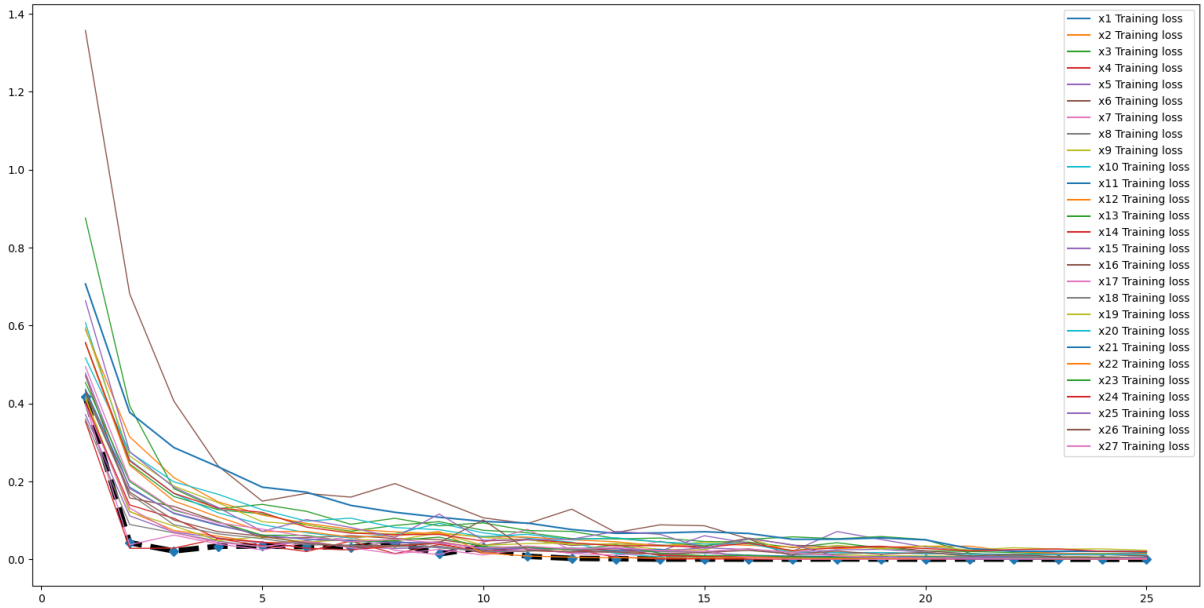


Slika 5.2.: Prikaz konvergiranja funkcija gubitka bez izvršavanja dropout() metode

Slike 5.3 i 5.4 prikazuju konvergiranje funkcije točnosti, odnosno gubitka prema zadovoljavajućem rezultatu od preko 98% za funkciju točnosti i 2% za funkciju gubitka. Modeli sa slika 5.3 i 5.4 su trenirani na arhitekturi s izvršavanjem metode *dropout()*, a model koji je odabran kao model s najboljim konvergiranjem k željenom rezultatu podebljan je crnom bojom.

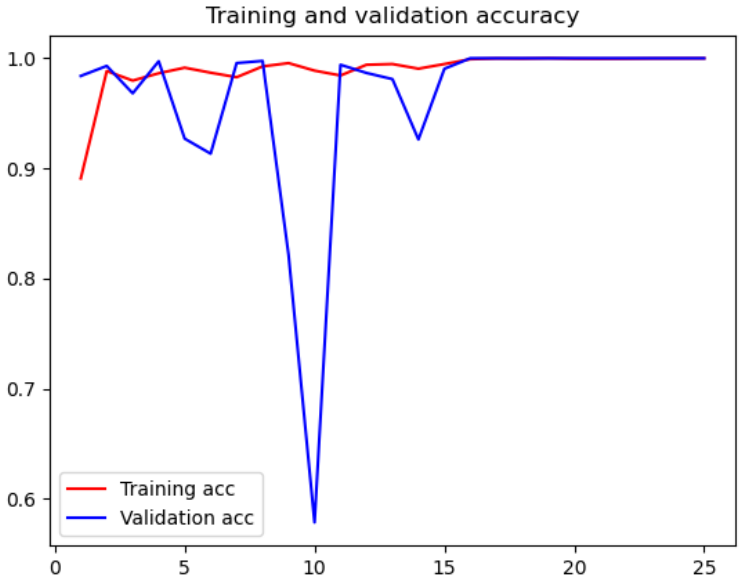


Slika 5.3.: Prikaz konvergiranja funkcija točnosti s izvršavanjem dropout() metode

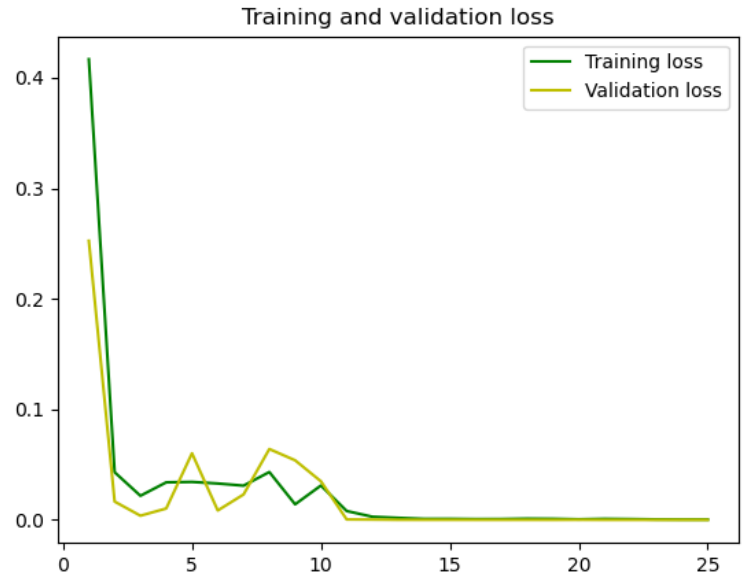


Slika 5.4.: Prikaz konvergiranja funkcija gubitka s izvršavanjem dropout() metode

Najbolje odabrani model koji je treniran na arhitekturi s izvršavanjem *dropout()* metode ima jedan ulazni sloj, jedan konvolucijski sloj te nema niti jedan potpuno povezani sloj te po svakom postojećem sloju ima 32 neurona. Slika 5.5 prikazuje funkcije točnosti na trening i validacijskom skupu, najboljeg modela koji je treniran na arhitekturi s izvršavanjem *dropout()* metode. Slika 5.6 prikazuje funkcije gubitka na trening i validacijskom skupu, najboljeg modela koji je treniran na arhitekturi s izvršavanjem *dropout()* metode.

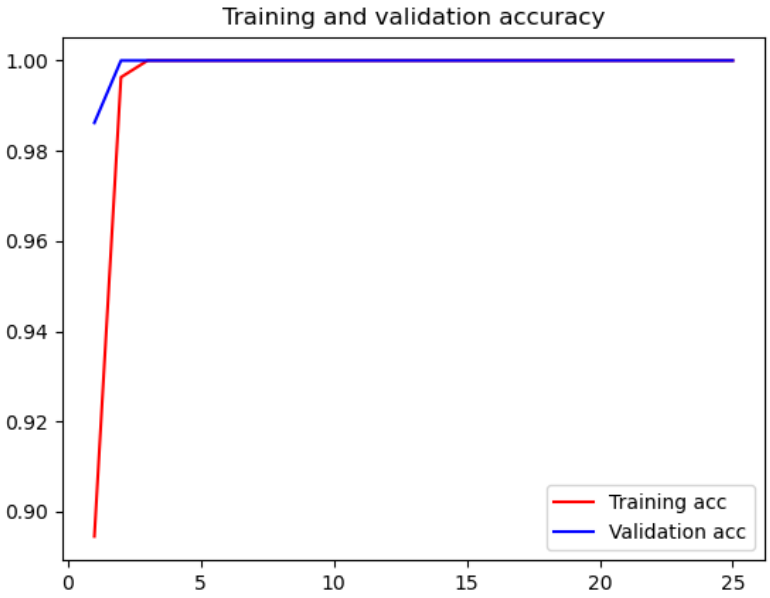


Slika 5.5.: Prikaz funkcija točnosti na trening i validacijskim skupovima s izvršavanjem dropout() metode

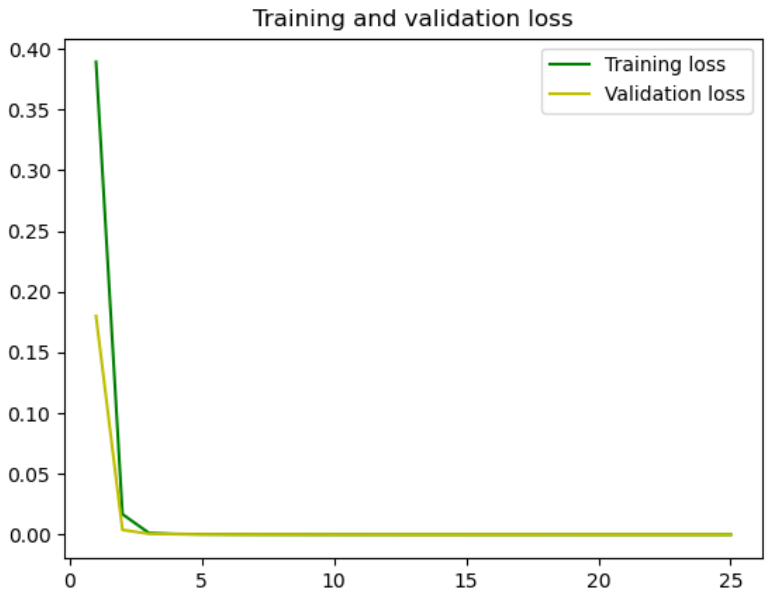


Slika 5.6.: Prikaz funkcija gubitkai na trening i validacijskim skupovima s izvršavanjem dropout() metode

Najbolje odabrani model koji je treniran na arhitekturi bez izvršavanja *dropout()* metode ima jedan ulazni sloj, jedan konvolucijski sloj te nema niti jedan potpuno povezani sloj te po svakom postojećem sloju ima 128 neurona. Slika 5.7 prikazuje funkcije točnosti na trening i validacijskom skupu, najboljeg modela koji je treniran na arhitekturi bez izvršavanja *dropout()* metode. Slika 5.8 prikazuje funkcije gubitka na trening i validacijskom skupu, najboljeg modela koji je treniran na arhitekturi bez izvršavanja *dropout()* metode.



Slika 5.7.: Prikaz funkcija točnosti na trening i validacijskim skupovima bez izvršavanja dropout() metode



Slika 5.8.: Prikaz funkcija gubitkai na trening i validacijskim skupovima bez izvršavanja dropout() metode

Odabrana dva najbolja modela iz obje skupine koristila su se kao testni modeli za predviđanje novog testnog skupa slika s bolestima vinove loze. Slike u novom testnom skupu se po prvi puta pojavljuju i ne nalaze se u trening skupu niti validacijskom skupu koji su korišteni pri treniranju modela konvolucijskih neuronskih mreža. Za predviđanje kategorija bolesti novog testnog skupa slika koristi se metoda *predict_classes()* koja vraća brojeve 0, 1, 2 ili 3 kao rezultat predviđanja određene kategorije gdje svaki broj predstavlja određenu kategoriju. Za procjenu kvalitete i uspješnosti treniranih modela koristi se matrica konfuzije. Matrica konfuzije prikazuje količinu predviđenih podataka te odnos između stvarnih podataka i predviđenih podataka. Matrica konfuzije daje četiri informacije koje se odnose na relaciju između stvarnih rezultata i predviđenih rezultata, a oni mogu biti:

- Istinito negativni (eng. *true negatives,* skraćeno TN)
- Lažno negativni (eng. *false negatives,* skraćeno FN)
- Istinito pozitivni (eng. *true positives,* skraćeno TP)
- Lažno pozitivni (eng. *false positives,* skraćeno FP*).*

Na temelju dobivenih informacija od matrice konfuzije mogu se dobiti relevantniji procjenitelji uspješnosti treniranog modela a to su:

- Preciznost (eng. *Precision*): daje informaciju koliko od pozitivno predviđenih podataka su istinito pozitivni, a računa se prema formuli (5-1).

$$Preciznost = \frac{TP}{FP+TP} \quad (5\text{-}1)$$

- Točnost (eng. *Accuracy):* daje informaciju od sveukupnih predviđenih podataka, koliko su točno predviđeni, a računa se prema formuli (5-2).

$$Točnost = \frac{TP+TN}{TP+TN+FP+FN} \quad (5\text{-}2)$$

- Odaziv (eng. *Recall*): daje informaciju koliko od pozitivnog predviđanja je istinito i točno predviđeno, a računa se prema formuli (5-3).

$$Odaziv = \frac{TP}{FN+TP} \quad (5\text{-}3)$$

- F1-ocjena (eng. F1-score): daje informaciju o odnosu između preciznosti i odaziva. Kada se želi postići ravnoteža odaziva i preciznosti, cilja se na veću F1-ocjenu, a ona se računa prema formuli (5-4).

$$F1 - ocjena = 2 * \frac{Preciznost*Odaziv}{Preciznost+Odaziv} \quad (5\text{-}4)$$

Matrica konfuzije dobije se kao povratni rezultat metode *confusion_matrix()* koja pripada klasi *math, tensorflow* biblioteke. Funkciji *confusion_matrix()* se kao parametri predaju stvarne kategorije i predviđene kategorije. Tablica 5.1 prikazuje matricu konfuzije dobivenu od modela treniranog s izvršavanjem *dropout()* metode, a tablica 5.2 prikazuje matricu konfuzije dobivenu od modela treniranog bez izvršavanja *dropout()* metode.

Tablica 5.1.: Matrica konfuzije modela s izvršavanjem dropout() metode

79	15	2	4
13	86	0	1
4	0	95	1
4	6	0	90

Tablica 5.2.: Matrica konfuzije modela bez izvršavanjem dropout() metode

72	23	3	2
11	86	0	3
1	0	98	1
3	5	0	92

Na temelju procjenitelja uspješnosti za tablicu 5.1 dobiveni su sljedeći izračuni:

- Srednja vrijednost preciznosti svih kategorija iznosi: 87,77%
- Srednja vrijednost odaziva svih kategorija iznosi: 87,5%
- Vrijednost točnosti iznosi: 90,44%
- Vrijednost za F1-ocjenu iznosi: 87,63%.

Na temelju procjenitelja uspješnosti za za tablicu 5.2 dobiveni su sljedeći izračuni:

- Srednja vrijednost preciznosti svih kategorija iznosi: 87,28%
- Srednja vrijednost odaziva svih kategorija iznosi: 87%
- Vrijednost točnosti iznosi: 87%
- Vrijednost za F1-ocjenu iznosi: 87,14%.

Na temelju testiranja i dobivenih rezultata može se uvidjeti da su svih 54 kombinacija istreniranih konvolucijskih neuronskih mreža u obje ispitne grupe (s izvršavanjem i bez izvršavanja metode *dropout()*) postigle zadovoljavajuće rezultate. Međutim, radi lakšeg testiranja i prikaza rezultata, izdvojena su dva trenirana modela konvolucijskih neuronskih mreža koja su najbrže konvergirala optimalnim rezultatima. Na temelju rezultata procjenitelja uspješnosti modela koji proizlaze iz matrica konfuzije može se zaključiti da nema velikih odstupanja između izdvojena dva modela. Može se reći da uspoređivanjem procjenitelja usjpešnosti tablice 5.1 i tablice 5.2, model s izvršavanjem *dropout()* metode ima blago veću prednost naspram modela bez izvršavanja *dropout()* metode, jer su rezultati svih procjenitelja uspješnosti modela veći. Najveća razlika između ova dva modela se iskazuje u točnosti predviđenih podataka, a ona iznosi 3,44%

6. ZAKLJUČAK

Kako bi se dobila vrhunska kvaliteta vina potrebno je imati visoko kvalitetno grožđe. Potrebno je postići određene prakse tijekom godine kako bi se osigurali uvjeti za postizanje visoko kvalitetnog grožđa. Jedna od tih praksi je i zaštita vinove loze od bolesti. Potrebno je na vrijeme djelovati kako bi se prepoznala i spriječila bolest vinove loze. Takav proces je dugotrajan, skup i sklon je ljudskim pogreškama. Potrebno je zato izraditi sustav koji će zamijeniti čovjeka u takvom procesu rada. Ideja ovog diplomskog rada je izraditi sustav za prepoznavanje bolesti vinove loze sa slika. Prvenstveno, važno je poznavati glavne principe računalnog vida i strojnog učenja kako bi se uspješno primijenili u daljnjim koracima pri izgradnji sustava za prepoznavanje bolesti vinove loze. Također potrebno je poznavati određene programske alate i programske jezike nad kojima bi se primijenilo stečeno znanje računalnog vida i strojnog učenja. Zbog jednostavnosti rada s bibliotekama za duboko učenje, primijenjen je programski jezik Python. Prije same izrade konvolucijske neuronske mreže i treniranja iste potrebno je pribaviti podatkovni skup slika i predobraditi ih u pogodan format za treniranje neuronskih mreža. Predobrada podataka podrazumijeva smanjivanje dimenzionalnosti slika, pretvaranje višekanalne slike u boji u monokromatsku, miješanje podataka i regularizacija podataka. Nadalje, bilo je potrebno izgraditi arhitekturu konvolucijske neuronske mreže tako da se na predobrađenim podacima istrenira više varijacija jedne konvolucijske neuronske mreže kako bi se moglo doći do zaključka koja varijacija konvolucijske neuronske mreže daje najbolje rezultate. Rezultati su pokazali kako sve dobivene kombinacije istreniranih modela konvolucijskih neuronski mreža postižu optimalne vrijednosti. Naposljetku, uočeno je kako konvolucijske neuronske mreže daju bolje rezultate od klasičnih metoda strojnog učenja, te su pogodne za prepoznavanje objekata sa slika. Glavne mane konvolucijskih neuronskih mreža su: zahtijevanje velikih podatkovnih skupova nad kojima će se trenirati te zahtijevanje velike procesorske moći. Ovaj diplomski rad bi se mogao unaprijediti tako da se primijene metode segmentacije slike za slučaj ako se na slici nalazi više različitih bolesti vinove loze.

LITERATURA

[1] B. D. Bašić, M. Čupić, J. Šnajder, Umjetne neuronske mreže, diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb, svibanj 2008., dostupno na: https://www.fer.hr/_download/repository/UmjetneNeuronskeMreze.pdf, [23.9.2020.]

[2] W. McCulloch, W. Pitts, *A Logical Calculus of Ideas Immanent in Nervous Activity*, 1943.

[3] Gulli, S. Pal, *Deep Learning with Keras*, *Pact Publishing Ltd.*, Birmingham, travanj 2017.

[4] *PyCharm* [online], dostupno na : https://www.jetbrains.com/pycharm/, [23.9.2020.]

[5] H. Zhu, Q. Liu, Y. Qi, X. Huang, F. Jiang, S. Zhang, *Plant identification based on very deep convolutional neural networks, Multimedia Tools and Applications*, br. 22, sv. 77, studeni 2018.

[6] Python [online] , wikipedia.org, dostupno na: https://en.wikipedia.org/wiki/Python_(programming_language), [23.9.2020.]

[7] OpenCV [online], opencv.org, dostupno na: https://opencv.org/about/, [23.9.2020.]

[8] G. Li, Z. Ma, H. Wang, *Image recognition of grape downy mildew and grape powdery mildew based on support vector machine, Proc international conference on computer and computing technologies in agriculture*. str. 151-162, Peking, Kina, 2012.

[9] G. Athanikar, P. Badar, *Potato leaf diseases detection and classification system,International Journal of Computer Science and Mobile Computing*, br. 5, sv. 2, str. 76-88, veljača 2016.

[10] S.H. Lee, C.S. Chan, P. Wilkin, P. Remagnino, *Deep-plant: plant identification with convolutional neural networks, IEEE international conference on image processing*, Quebec, Kanada 2015,

[11] S. Mohanty, D. Hughes, M. Salathé, *Using deep learning for image-based plant disease detection,Frontiers in Plant Science, br.* 7 , str. 1419, Rujan 2016.

[12] A. Krizhevsky, I. Sutskever, G. Hinton, *ImageNet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems*, br. 25, str. 1106-1114, 2016.

[13] Simonyan K, Zisserman A., *Very deep convolutional networks for large-scale image recognition*., *International Conference on Learning Representations*, 2014., dostupno na : https://arxiv.org/abs/1409.1556 [23.9.2020.]

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.Going deeper with convolutions, IEEE conference on computer vision and pattern recognition.*, Columbus, USA, 2014, dostupno na : https://arxiv.org/abs/1409.4842 [23.9.2020.]

[15] Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer, K. Densenet: *Implementing efficient convnet descriptor pyramids*. 2014. dostupno na: https://arxiv.org/abs/1404.1869v1. [23.9.2020.]

[16] K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition,IEEE conference on computer vision and pattern recognition.*, str. 770-778, Boston, Massachusetts , 2016, dostupno na: https://arxiv.org/abs/1512.03385 [23.9.2020.]

[17] Szegedy C, Ioffe S, Vanhoucke V, Alemi A, *Inception-v4, inception-resnet and the impact of residual connections on learning*, dostupno na : https://arxiv.org/abs/1602.07261 [23.9.2020.]

[18] Lin M, Chen Q, Yan S, *Network in network*, dostupno na: https://arxiv.org/abs/1312.4400 [23.9.2020.]

[19] T. Birka, Implementacija neuronske mreže na Zybo razvojnom sustavu, diplomski rad, Fakultet elektrotehnike računarstva i informacijskih tehnologija, Osijek, Rujan 2019., dostupno na : https://zir.nsk.hr/islandora/object/etfos%3A2568, [24.9.2020.]

[20] S. Dumančić, Neuronske mreže, diplomski rad, Odjel za fiziku, Osijek,Listopad 2014., dostupno na :http://www.mathos.unios.hr/~mdjumic/uploads/diplomski/dum05.pdf, [24.9.2020.]

[21] V. Vukotić, Raspoznavanje objekata dubokim neuronskim mrežama, diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb, lipanj 2014., dostupno na: http://www.zemris.fer.hr/~ssegvic/project/pubs/vukotic14ms.pdf, [24.9.2020.]

[22] *TensorFlow* [online], wikipedia.org, dostupno na: https://en.wikipedia.org/wiki/TensorFlow, [23.9.2020.]

[23] *Keras Documentation* [online], dostupno na: https://keras.io/, [24.9.2020.]

[24] *Matplotlib* [online], dostupno na : https://matplotlib.org/3.2.1/users/history.html, [24.9.2020.]

[25] *NumPy* [online], dostupno na https://numpy.org/doc/stable/about.html , [24.9.2020.]

[26] *Random* [online], dostupno na :https://docs.python.org/3/library/random.html [24.9.2020.]

[27] *Pickle* [online], dostupno na :https://docs.python.org/3/library/pickle.html [24.9.2020.]

[28] *Time* [online], dostupno na :https://realpython.com/python-time-module/# [24.9.2020.]

[29] D. Ciresan, U. Meier, J. Masci, i J. Schmidhuber, *Multi-column deep neural network for traffic sign classification.*, *Neural Networks*, br. 32, str. 333–338, 2012.

SAŽETAK

Za konkurentnost na tržištu u proizvodnji vina potrebno je imati najkvalitetnije moguće sirovine grožđa kako bi se proizvelo što kvalitetnije vino. Vinari u modernom dobu izdvajaju velika novčana sredstva za obranu vinograda od bolesti vinove loze. Utrošena novčana sredstva se značajnije smanjuju ranim otkrivanjem bolesti vinove loze. Zadatak ovog diplomskog rada bio je istražiti i izraditi sustav za prepoznavanje bolesti vinove loze sa slika, a pri tome su korištene metode neuronskih mreža. Prije samog treniranja neuronskih mreža, potrebno je prikupiti i predobraditi podatke. Programska podrška za ovakav sustav napisana je u programskom jeziku Python. Nadalje, dobiveni rezultati treniranih modela su uspoređeni te su odabrani najbolji kojima su se vršila testiranja. Neuronske mreže su se pokazale kao vrlo dobra metoda za uspješno prepoznavanje bolesti vinove loze.

Ključne riječi: duboko učenje, neuronske mreže, prepoznavanje bolesti, Python, vinova loza

ABSTRACT

Title: Recognition of grape vine diseases from images

In order to be competitive on the market, in wine production it is necessary to have grapes of the highest possible quality in order to produce wine of the highest quality. Winemakers in the modern age give large sums of money to defend vineyards from diseases. By early detection of diseases, the need for spending funds on protection is significantly reduced. The aim of this study was to investigate and develop a system for detection of grape vine diseases from images, using neural network methods. Before training of neural networks, it is necessary to collect and pre-process the data. Software for such system is written in the Python programming language. Furthermore, the obtained results of the trained models were compared and the best ones were selected and tested. Neural networks have proven to be a very good method for successfully recognizing grape vine diseases.

Key words: deep learning, neural networks, disease recognition, Python, grape vine

ŽIVOTOPIS

Krešimir Markota rođen je 5. Travnja 1995. godine u Požegi. Završio je Osnovnu školu fra Kaje Adžića u Pleternici, nakon čega upisuje Gimnaziju u Požegi, smjer Prirodoslovno – matematički, koju završava 2014. godine. Iste godine upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI

P.1 Direktorij s Python skriptama nalazi se na CD-u

P.2 Verzija diplomskog rada u .pdf formatu nalazi se na CD-u

p.3 Verzija diplomskog rada u .docx formatu nalazi se na CD-u