SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

IZRADA ANDROID APLIKACIJE ZA KLASIFIKACIJU BILJAKA I DRVEĆA NA TEMELJU SLIKA

Goran Bogetić

Split, kolovoz 2025.

|  |  |  |
| --- | --- | --- |
|  | SVEUČILIŠTE U SPLITU  FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE |  |

Sveučilišni diplomski studij: **Naziv studija**

Smjer/Usmjerenje: **Naziv smjera/usmjerenja**

Oznaka programa: 210

Akademska godina: 2023./2024.

Ime i prezime: **IME PREZIME**

JMBAG: xxxxxxxxxx

**ZADATAK DIPLOMSKOG RADA**

Naslov: **NASLOV RADA**

Zadatak: Prilikom prijave diplomskog rada mentor definira temu i radni naslov, a točan naslov rada utvrđuje se po završetku rada. Diplomski se rad, potpisan od strane mentora, predaje Odboru za diplomski rad u tri tiskana primjerka s tvrdim uvezom te u elektroničkoj formi. Izradu diplomskog rada ocjenjuje mentor, a predsjednik Odbora za diplomski rad potvrđuje zadatak i konačni naslov rada. Ukoliko je predsjednik Odbora za diplomski rad ujedno i mentor, diplomski rad uz mentora potpisuje i jedan od članova Odbora. **Ovu stranicu piše mentor prije predaje rada nakon što je utvđen konačni naslov rada.**

Rad predan:

|  |  |  |
| --- | --- | --- |
| Predsjednik  Odbora za diplomski rad: |  | Mentor: |
|  |  |  |
| prof. dr. sc. Ime Prezime |  | izv. prof. dr. sc. Ime Prezime |

**IZJAVA**

Ovom izjavom potvrđujem da sam diplomski rad s naslovom IZRADA ANDROID APLIKACIJE ZA KLASIFIKACIJU BILJAKA I DRVEĆA NA TEMELJU SLIKA pod mentorstvom dr. sc. Josipa Vasilja pisao samostalno, primijenivši znanja i vještine stečene tijekom studiranja na Fakultetu elektrotehnike, strojarstva i brodogradnje, kao i metodologiju znanstveno-istraživačkog rada, te uz korištenje literature koja je navedena u radu. Spoznaje, stavove, zaključke, teorije i zakonitosti drugih autora koje sam izravno ili parafrazirajući naveo/la u diplomskom radu citirao/la sam i povezao/la s korištenim bibliografskim jedinicama.

Student/ica

Goran Bogetić

**SADRŽAJ**

[1 UVOD 1](#_Toc207223179)

[2 TEORIJSKA POZADINA I POVEZANA ISTRŽIVANJA 2](#_Toc207223180)

[2.1 Struktura diplomskog rada – obvezni dijelovi radnje 2](#_Toc207223181)

[2.1.1 Osnovni koncepti 2](#_Toc207223182)

[2.1.2 Usporedba koncepata 5](#_Toc207223183)

[2.2 Neuronske mreže i duboko učenje 6](#_Toc207223184)

[2.2.1 Konvolucijske neuronske mreže 7](#_Toc207223185)

[2.2.2 Transfer learning i fine-tuning 8](#_Toc207223186)

[2.3 Računalni vid 10](#_Toc207223187)

[2.4 Povezna istraživanja i postojeći sustavi 12](#_Toc207223188)

[3 IZRADA BAZE PODATAKA 13](#_Toc207223189)

[3.1 Ciljevi i dizajn baze podataka 13](#_Toc207223190)

[3.2 Izvori podataka 15](#_Toc207223191)

[3.3 Tehnička izvedba prikupljanja slika 17](#_Toc207223192)

[3.4 Organizacija i podjela podataka 23](#_Toc207223193)

[3.5 Napomene za pisanje nekih dijelova radnje 24](#_Toc207223194)

[3.6 O temi/problemu 24](#_Toc207223195)

[3.7 O strukturi radnje 26](#_Toc207223196)

[4 PLAN KOMUNIKACIJE 30](#_Toc207223197)

[4.1 Komunikacija na projektu 30](#_Toc207223198)

[4.2 Mail-adrese i telefonski brojevi 30](#_Toc207223199)

[4.3 Nazivi dokumenata 30](#_Toc207223200)

[4.4 Pitanja 31](#_Toc207223201)

[5 PLANIRANJE I EVIDENCIJA 32](#_Toc207223202)

[5.1 IT-tvrtke u Hrvatskoj 32](#_Toc207223203)

[5.2 Planiranje aktivnosti 32](#_Toc207223204)

[5.3 Označavanje slika, tablica i ostalih objekata 33](#_Toc207223205)

[5.4 Evidencija aktivnosti 35](#_Toc207223206)

[6 KONTROLA I NADZOR DIPLOMSKOG 37](#_Toc207223207)

[7 METODA RADA I PISANJA RADNJE 38](#_Toc207223208)

[7.1 Pisanje radnje 38](#_Toc207223209)

[7.2 ECTS 38](#_Toc207223210)

[7.3 Alati za pisanje 39](#_Toc207223211)

[7.4 Oblik finalnog dokumenta (obvezno!) 39](#_Toc207223212)

[8 DOGOVOR OKO POČETNIH OBRAZACA ZA PISANJE RADNJE 40](#_Toc207223213)

[8.1 Neka to bude ovaj dokument. (heading 2) 40](#_Toc207223214)

[8.1.1 On sadrži potrebne stilove. (heading 3) 40](#_Toc207223215)

[8.1.2 Upute za napredno korištenje Worda 41](#_Toc207223216)

[9 NAPOMENE 42](#_Toc207223217)

[10 NAPOMENE PROFESORA 43](#_Toc207223218)

[11 UPUTE ZA PREZENTACIJU 44](#_Toc207223219)

[12 UNOS RADNJE U BAZE PODATAKA I PDF-PRINT 46](#_Toc207223220)

[12.1 Baze podataka 46](#_Toc207223221)

[12.2 PDF-printanje radnje 47](#_Toc207223222)

[13 UPUTE ZA LITERATURU 48](#_Toc207223223)

[13.1 Korisni izvori literature (osim Wikipedije) 48](#_Toc207223224)

[13.2 SWEBOK 48](#_Toc207223225)

[13.2.1 Način pretraživanja 48](#_Toc207223226)

[13.2.2 Navođenje izvora literature 49](#_Toc207223227)

[13.2.3 Literatura za mentora 50](#_Toc207223228)

[14 ZAKLJUČAK 51](#_Toc207223229)

[LITERATURA 52](#_Toc207223230)

[PRILOZI 54](#_Toc207223231)

[Kazalo slika, tablica i kodova 54](#_Toc207223232)

[Kazalo slika 54](#_Toc207223233)

[Kazalo tablica 54](#_Toc207223234)

[Kazalo kodova 55](#_Toc207223235)

[Popis oznaka i kratica 55](#_Toc207223236)

[Ostali prilozi i dokumentacija 55](#_Toc207223237)

[Kazalo profesorovih napomena 55](#_Toc207223238)

[Upravljanje izmjenama 55](#_Toc207223239)

[SAŽETAK/ABSTRACT I KLJUČNE RIJEČI/KEYWORDS 59](#_Toc207223240)

# UVOD

Umjetna inteligencija postala je jedna od najznačajnijih tehnologija 21. stoljeća, s utjecajem na gotovo svaki aspekt društva – od financija i zdravstva do obrazovanja i poljoprivrede. Najveću primjenu pronašla je u analizi velikih količina podataka, automatiziranom donošenju odluka te inteligentnom rješavanju složenih problema iz stvarnog svijeta. Posebno su strojno učenje i duboko učenje omogućili značajan napredak u obradi prirodnog jezika i računalnoj obradi slika, čime je otvoren put sustavima koji mogu obavljati zadatke do tada rezervirane za stručnjake, uz sve veću preciznost koja se kontinuirano poboljšava.

Za ovaj rad posebno je važan računalni vid, odnosno sposobnost automatskog prepoznavanja i klasifikacije objekata na slikama. On nalazi široku primjenu, od medicinske dijagnostike i autonomne vožnje do praćenja okoliša i sigurnosnih sustava. Kao rezultat toga, računalni vid postao je moćan alat u situacijama gdje su ručno promatranje i klasifikacija vremenski zahtjevni, skloni pogreškama ili nepraktični u velikim razmjerima. Točna identifikacija biljnih vrsta ključna je u poljoprivredi, šumarstvu i ekologiji, no često zahtijeva stručno botaničko znanje koje nije lako dostupno široj javnosti. S razvojem aplikacija temeljenih na umjetnoj inteligenciji, danas je moguće premostiti te prepreke pružanjem automatiziranih i jednostavnih alata koji identifikaciju biljaka čine dostupnijom i bržom.

U ovom radu prikazat će se postupak izrade Android aplikacije za automatsku klasifikaciju biljaka i drveća, u slučaju kada je potrebno razviti vlastitu bazu podataka te na temelju nje istrenirati model za prepoznavanje. U drugom poglavlju bit će objašnjena teorijska osnova strojnog učenja i princip rada algoritama temeljenih na neuronskim mrežama. Treće poglavlje bavit će se postupkom izrade baze podataka od početka, dok će četvrto prikazati treniranje modela. Peto poglavlje posvećeno je izradi Android aplikacije koja implementira automatsku klasifikaciju na temelju slika korištenjem prethodno istreniranog modela.

# TEORIJSKA POZADINA I POVEZANA ISTRŽIVANJA

Ovo poglavlje pružiti će teorijsku pozadinu i osnovne koncepte potrebne za razumijevanje razvoja predložene Android aplikacije za automatsku klasifikaciju biljaka i drveća. Uvesti će se temeljni principi umjetne inteligencije i strojnog učenja, s posebnim naglaskom na nadzirano strojno učenje koje je vrlo važno u klasifikaciji slika.

Nakon toga prelazi se na neuronske mreže i duboko učenje uz naglasak na arhitekture koje se najčešće koriste za vizualno prepoznavanje, poput konvolucijskih neuronskih mreža. Zatim će biti pokazano područje računalnog vida, gdje će biti pokriveni osnovni procesi prepoznavanja slika i njegove raznolike primjene. Naposljetku, biti će pregled povezanih istraživanja i postojećih mobilnih aplikacija za identifikaciju biljaka, trenutnim ograničenjima i motivaciju za ovaj rad.

## Struktura diplomskog rada – obvezni dijelovi radnje

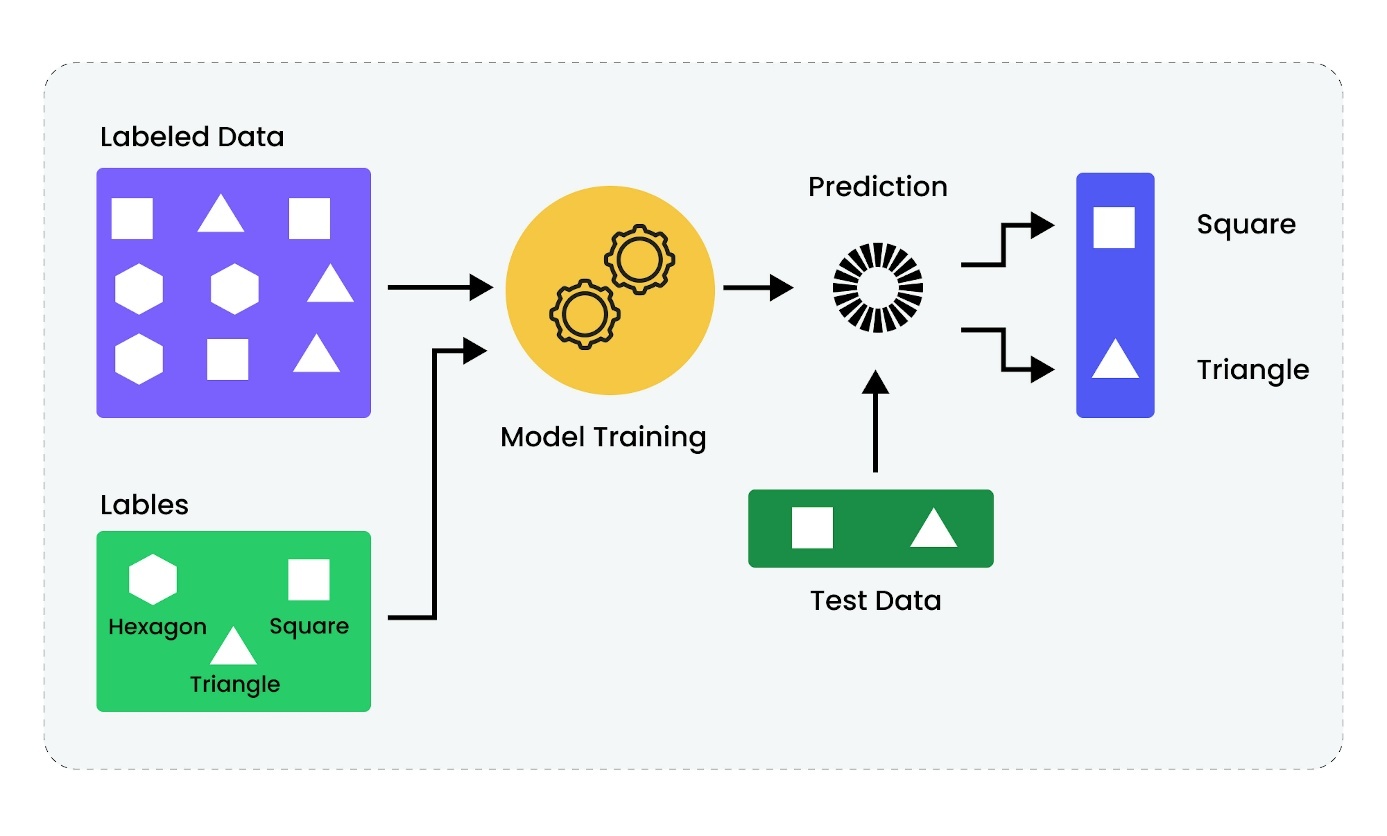
Umjetna inteligencija odnosi se na sposobnost strojeva za izvršavanje radnji obično povezanih sa ljudskom inteligencijom, poput rasuđivanja, percepcije, učenja te donošenja odluka. Unutar ovog vrlo širokog područja, strojno učenja omogućava tim sustavima poboljšanje performansi u zadatcima učenjem iz podataka umjesto da se eksplicitno programiraju pravila određene domene [1].

Osnovni cilj strojnog učenja jest generalizacija, odnosno sposobnost modela da nakon učenja na nekom skupu podataka ispravno prepoznaje obrasce i donosi točne zaključke o novim obrascima koje nikada prije nije vidio. Proces učenja odvija se kroz treniranje modela, pri čemu se parametri modela postupno prilagođavaju u cilju minimiziranja pogrešaka.

### Osnovni koncepti

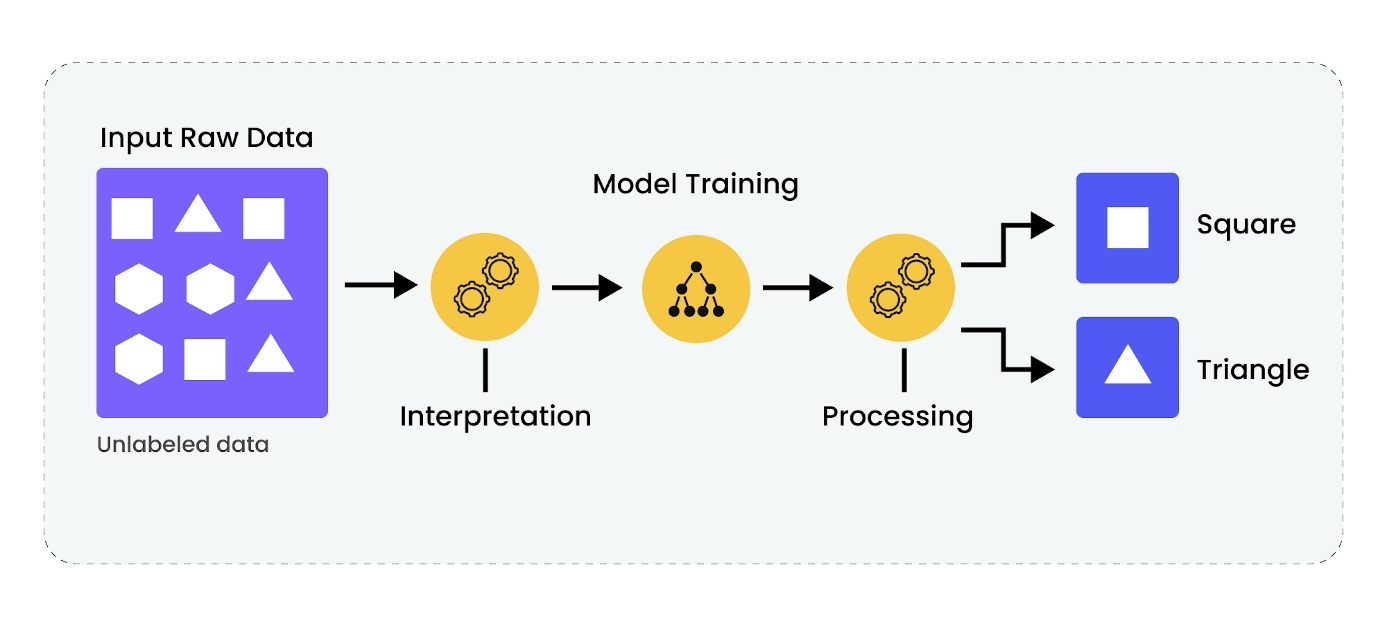
Strojno učenja ima više pristupa, pri čemu su sljedeća najvažnija, nadzirano i nenadzirano strojno učenje, dok se u suvremenim primjenama sve češće koristi i podržano strojno učenje.

Nadzirano strojno učenja podrazumijeva korištenje označenih podataka prilikom treniranja, odnosno korištenje podataka u kojima su uz ulazne podatke poznati i željeni izlazni podatci. Model uči povezati ulaze s pripadajućim izlazima, kao što je vidljivo na slici Slika 2‑1 te se najčešće koristi u zadatcima klasifikacije, npr. prepoznavanje vrste stabla prema slici lista, i regresije, npr. procjena veličine lista. Prednost nadziranog strojnog učenja jest velika preciznost ukoliko se podatci kvalitetno i točno označeni, dok je glavni nedostatak velik trošak i vrijeme potrebno za ručno označavanje velikih skupova podataka [2].



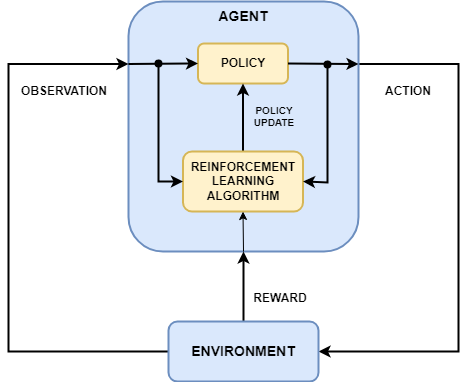
Slika 2‑1 Prikaz kako radi nadzirano strojno učenje. (preuzeto s <https://maddevs.io/blog/semi-supervised-learning-explained/> )

Nenadzirano strojno učenje, za razliku od nadziranog učenja, ne koristi označene podatke, već algoritmi sami otkrivaju skrivene strukture i obrasce u podatcima, a prikaz kako radi nenadzirano strojno učenje može se vidjeti na slici Slika 2‑2. Najčešće se klasteriranja, što je grupiranje sličnih primjera, i smanjenje dimenzionalnosti, gdje se veliki broj varijabli u podacima svodi na manji broj, pri čemu se nastoji zadržati što je moguće više informacija. Glavna prednost ovog pristupa je mogućnost pronalaska nepoznatih odnosa u podacima, ali evaluacija i interpretacija rezultata često su složenije nego kod nadziranog učenja [3].



Slika 2‑2 Prikaz kako radi nenadzirano strojno učenje (preuzeto s <https://maddevs.io/blog/semi-supervised-learning-explained/>)

Podržano strojno učenje predstavlja dodatan koncept , gdje sustav uči putem interakcije s okolinom, primajući nagrade ili kazne na temelju svojih akcija, kao što se može vidjeti na slici Slika 2‑3. Iako se rjeđe koristi u zadacima klasifikacije slika, važan je u područjima poput robotike i autonomnih vozila [4].



Slika 2‑3 Prikaz podržanog strojnog učenja (preuzeto s <https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html>)

### Usporedba koncepata

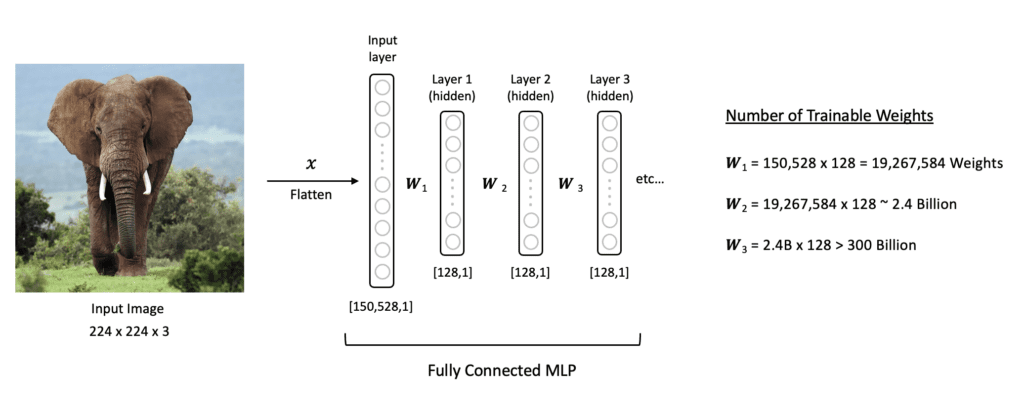
U tablici Tablica 1 može se vidjeti direktna usporedba tri prethodno spomenuta koncepta.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nadzirano strojno učenj** | **Nenadzirano strojno učenje** | **Podržano strojno učenje** |
| **Ulazni podaci** | Označeni (ulaz i izlaz) | Neoznačeni (samo ulaz) | Stanja i akcije, bez oznaka |
| **Cilj** | Predviđanje poznatih izlaza | Otkrivanje strukture i uzoraka | Maksimizacija kumulativne nagrade |
| **Tipični zadaci** | Klasifikacija, regresija | Klasteriranje, smanjenje dimenzionalnosti | Upravljanje, donošenje odluka |
| **Evaluacija** | Točnost, preciznost, odziv, F1-mjera | Teže mjerljiva, često subjektivna | Vrijednost nagrade kroz epizode |
| **Prednost** | Visoka preciznost uz kvalitetne oznake | Otkiva nepoznate obrasce | Uči iz iskustva i interakcije |
| **Nedostatak** | Skupo i zahtjevno označavanje podataka | Rezultati teško interpretabilni | Velika složenost i potreba za simulacijom |

Tablica 1 Usporedba između nadziranog, nenadziranog i podržanog strojnog učenja

## Neuronske mreže i duboko učenje

Neuronske mreže predstavljaju temeljnu klasu algoritama strojnog učenja inspiriranih radom ljudskog mozga. Osnovni element svake neuronske mreže jest umjetni neuron, koji prima ulazne podatke, postavlja im težine, primjenjuje funkciju aktivacije te generira izlaz. Perceptron bio je najraniji oblik neuronske mreže sposoban za rješavanje jednostavnih problema linearne separabilnosti [5]. Kasnije su uvedeni višeslojni perceptroni, tzv. MLP, koji preko skrivenih slojeva omogućuju modeliranje složenijih nelinearnih odnosa, a primjer takvog MLP-a može se vidjeti na slici Slika 2‑4.



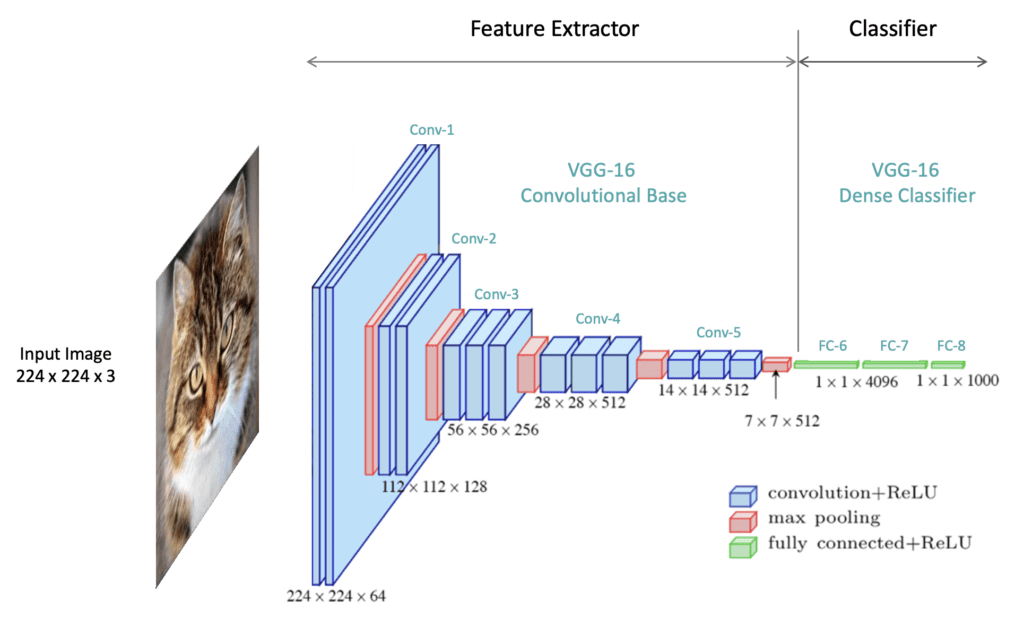
Slika 2‑4 Primjer izgleda višeslojnog perceptrona, tzv. MLP-a (preuzeto s <https://learnopencv.com/understanding-convolutional-neural-networks-cnn/>)

Razvoj računalne snage i dostupnost velike količine podataka doveli su do pojave dubokog učenja, što je temeljeno na dubokim neuronskim mrežama koje sadrže veliki broj slojeva. Duboke mreže sposobne su automatski učiti složene reprezentacije podataka, što ih čini posebno pogodnima za obradu slike, zvuka i prirodnog jezika [6].

### Konvolucijske neuronske mreže

U računalnom vidu posebno su značajne konvolucijske neuronske mreže (engl. Convolutional Neural Networks, CNN). Njihova ključna prednost u odnosu na prethodno spomenite MLP arhitekture jest u konvolucijskim slojevima, koji omogućuju automatsko izdvajanje značajki iz slika, npr. rubova, oblika, tekstura, kroz primjenu filtara. Tipična struktura CNN-a sastoji se od niza međusobno povezanih slojeva. Konvolucijski slojevi imaju ulogu automatskog izdvajanja značajki iz ulazne slike, poput rubova, oblika ili tekstura. Nakon njih slijede pooling slojevi, koji smanjuju dimenzionalnost podataka i povećavaju robusnost modela na translacije ili male promjene u slici. Na kraju mreže nalaze se potpuno povezani slojevi, koji na temelju prethodno izdvojenih značajki provode završnu klasifikaciju.

Razvoj CNN arhitektura značajno je unaprijedio računalni vid. Počevši od AlexNet-a [7], koji je donio proboj u prepoznavanju slika, preko VGGNet-a i ResNet-a koji su omogućili dublje i učinkovitije mreže, pa sve do mobilno optimiziranih modela poput MobileNet-a, CNN-ovi su postali temelj modernih sustava računalnog vida.

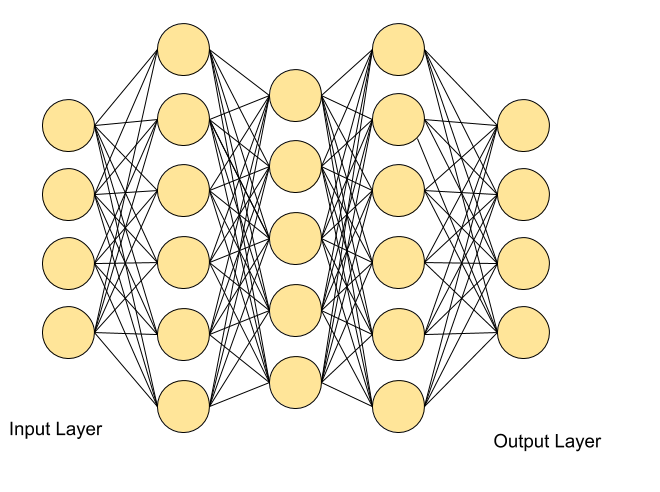


Slika 2‑5 Izgled konvolucijskih slojeva u VGG-16 konvolucijskoj neuralnoj mreži (preuzeto s <https://learnopencv.com/understanding-convolutional-neural-networks-cnn/>)

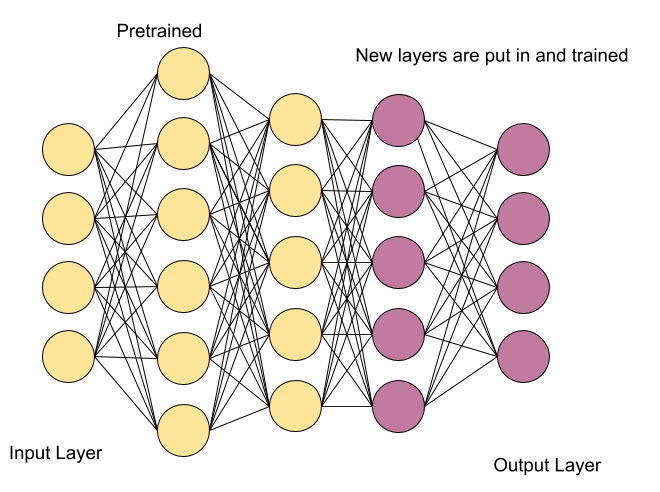
### Transfer learning i fine-tuning

Treniranje dubokih neuronskih mreža od početka zahtijeva velike količine podataka i značajne računalne resurse, što u mnogim slučajevima nije izvedivo. Zbog toga se u praksi često koristi tzv. transfer learning, tehnika kojom se unaprijed istrenirani modeli, najčešće na velikim skupovima podataka poput ImageNet-a, prilagođavaju novim zadacima. Uobičajeni pristup sastoji se od zamrzavanja ranijih slojeva, koji sadrže opće značajke poput rubova i osnovnih oblika, dok se treniraju samo završni slojevi. Alternativno, moguće je dodatno fino podešavati i dublje slojeve mreže, čime se postiže bolja specijalizacija za konkretni problem. Usporedba prikaze neuronske mreže trenirane od početka, slika Slika 2‑6, i napravljene tehnikom transfer learninga, slika Slika 2‑7.

Na ovaj način moguće je izgraditi kvalitetne modele i kada su dostupni manji skupovi podataka, primjerice u klasifikaciji biljnih vrsta ili detekciji bolesti na lišću. Transfer learning danas predstavlja jedan od ključnih pristupa u računalnom vidu te se široko primjenjuje i u akademskim istraživanjima i u industrijskim projektima.



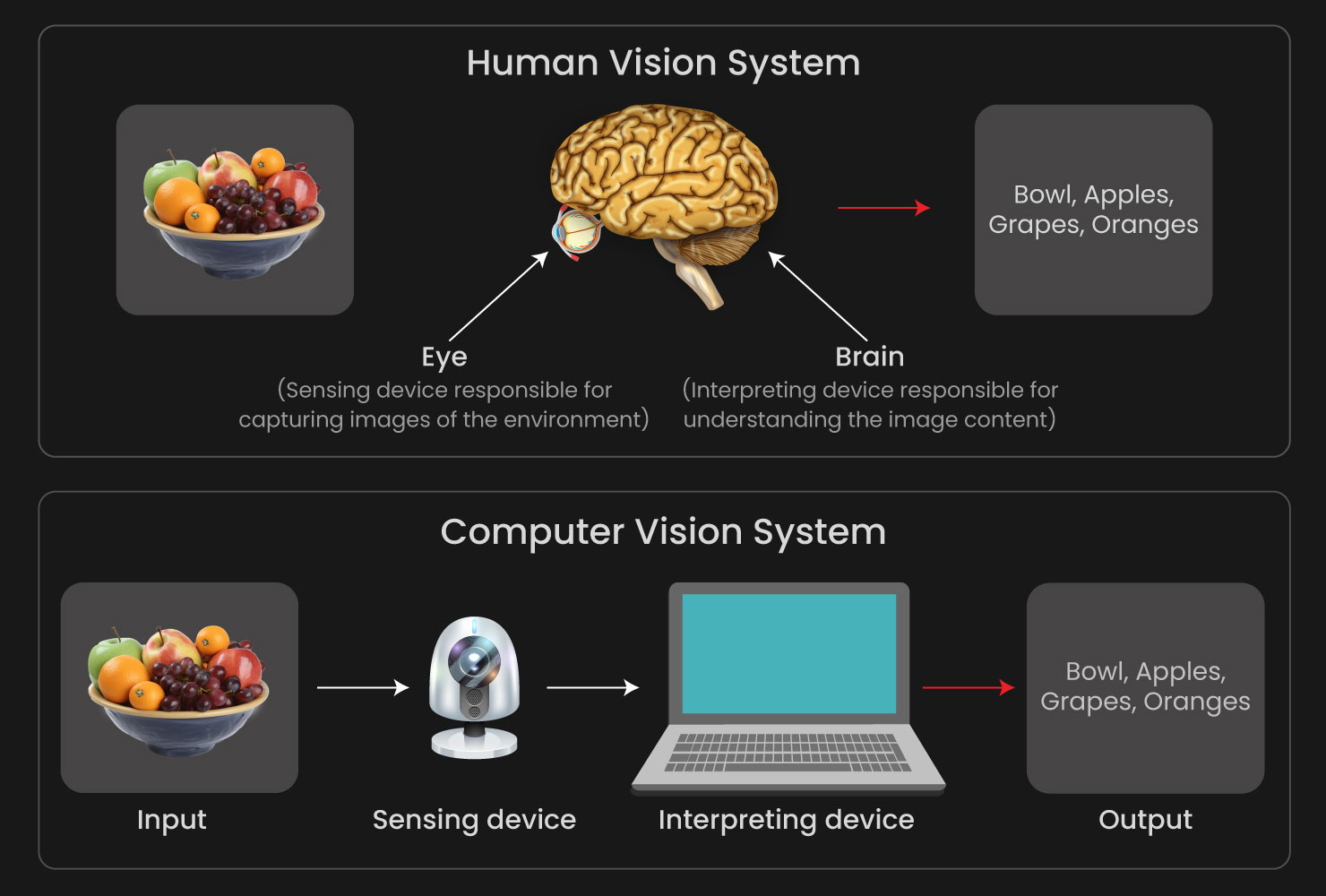
Slika 2‑6 Neuronska mreža trenirana od počeka (preuzeto s <https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective>)

****

Slika 2‑7 Neuronska mreža nakon upotrebe tranfer learning tehnike (preuzeto s <https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective>)

## Računalni vid

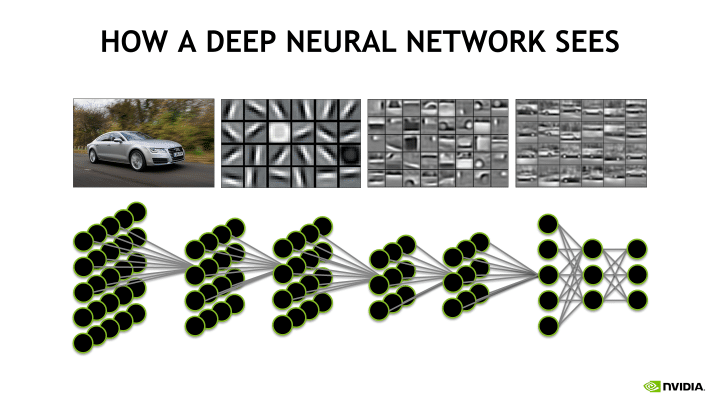
Računalni vid (engl. Computer Vision) predstavlja područje umjetne inteligencije koje se bavi omogućavanjem računalima da interpretiraju i razumiju vizualne informacije iz okoline. Glavni cilj računalnog vida jest omogućiti strojevima da, na temelju slika ili videozapisa, prepoznaju objekte, scene i obrasce te da na temelju tih informacija donose odluke ili izvode akcije. U osnovi, računalni vid nastoji replicirati ljudsku sposobnost vizualne percepcije, ali u digitalnom obliku, pri čemu se koristi snaga algoritama i procesorske arhitekture [8]. Ispod, na slici Slika 2‑8, se nalazi ilustracija koja prikazuje usporedbu između ljudskog vida i računalnog vida.



Slika 2‑8 Usporedba ljudskog vida i računalnog vida (preuzeto s <https://imagevision.ai/blog/understanding-computer-vision-a-technical-overview/>)

Proces računalnog vida obično započinje prikupljanjem vizualnih podataka, najčešće putem digitalnih kamera ili senzora. Slike se zatim podvrgavaju različitim tehnikama predobrade, koje uključuju prilagodbu veličine, normalizaciju boja, smanjivanje šuma ili povećanje kontrasta, s ciljem dobivanja kvalitetnih i konzistentnih ulaznih podataka [9]. Nakon toga slijedi faza ekstrakcije značajki, u kojoj algoritmi izdvajaju vizualne obrasce poput rubova, oblika ili tekstura, a u suvremenim pristupima te značajke uče se automatski korištenjem dubokih neuronskih mreža. Na slici Slika 2‑9 prikazano kako neuronska mreža prepoznaje značajke sa slika. Konačna faza jest klasifikacija ili detekcija objekata, u kojoj model donosi odluku kojoj klasi pripada određeni uzorak ili gdje se u slici nalazi željeni objekt.

Razvoj računalnog vida značajno je unaprijeđen pojavom dubokog učenja, a posebno konvolucijskih neuronskih mreža. One su omogućile automatsko učenje hijerarhijskih značajki iz podataka, čime je smanjena potreba za ručnim dizajnom algoritama i omogućeno postizanje rezultata koji u mnogim zadacima nadmašuju ljudsku razinu prepoznavanja. Osim toga, tehnike poput proširivanja podataka dodatno poboljšavaju učinkovitost modela stvaranjem varijacija iz postojećih slika, čime se povećava robusnost mreže i smanjuje rizik od pretreniranja.



Slika 2‑9 Prikaz kako neuronska mreža vidi različite značajke sa slike (preuzeto s <https://www.nvidia.com/en-gb/glossary/computer-vision/>)

Primjene računalnog vida vrlo su široke i obuhvaćaju područja od medicinske dijagnostike, gdje se koristi za otkrivanje tumora ili analizu radioloških snimaka, do autonomnih vozila koja zahtijevaju pouzdano prepoznavanje objekata u stvarnom vremenu. U poljoprivredi računalni vid omogućuje klasifikaciju biljnih vrsta, otkrivanje bolesti ili nadzor usjeva, dok je u ekologiji i šumarstvu važan alat za identifikaciju drveća, praćenje bioraznolikosti i nadzor okoliša [10]. Upravo zbog tih mogućnosti računalni vid predstavlja ključnu tehnologiju za razvoj sustava koji čine prepoznavanje i klasifikaciju biljaka dostupnijima i učinkovitijima široj javnosti.

## Povezna istraživanja i postojeći sustavi

Razvoj dubokog učenja i računalnog vida u posljednjem desetljeću potaknuo je niz istraživanja usmjerenih na automatsku identifikaciju biljnih vrsta. Brojna znanstvena djela pokazala su da konvolucijske neuronske mreže postižu vrlo dobre rezultate u zadacima klasifikacije biljaka i drveća, nadmašujući klasične metode koje su se oslanjale na ručno dizajnirane značajke poput oblika lista ili boje [11]. Na primjer, Hughes i Salathé (2015) koristili su CNN modele za detekciju biljnih bolesti na slikama lišća te postigli visoku razinu točnosti čak i u slučajevima kada su vizualne razlike između zdravih i oboljelih listova bile minimalne [12]. Slično tome, Lee i sur. (2017) demonstrirali su primjenjivost dubokog učenja na prepoznavanje različitih vrsta drveća na temelju fotografija krošnji, čime su potvrdili potencijal ove tehnologije u šumarstvu [13].

U području poljoprivrede i ekologije razvijene su i specijalizirane baze podataka poput PlantCLEF i PlantVillage, koje su postale standard za treniranje i evaluaciju modela u ovom području [14]. Ove baze sadrže stotine tisuća označenih slika biljaka i drveća, omogućujući istraživačima da testiraju različite arhitekture neuronskih mreža i uspoređuju njihove performanse.

Paralelno s akademskim istraživanjima, razvijen je i niz komercijalnih i otvorenih mobilnih aplikacija za identifikaciju biljaka. Među najpoznatijima ističu se PlantNet, PlantSnap i Google Lens. PlantNet se temelji na suradničkom prikupljanju podataka te omogućuje korisnicima da fotografijama doprinose rastu bazi slika i poboljšanju sustava [15]. PlantSnap koristi unaprijed istrenirane modele dubokog učenja za prepoznavanje više od 600.000 biljnih vrsta, dok Google Lens integrira prepoznavanje biljaka u širi sustav računalnog vida za različite objekte[16].

# IZRADA BAZE PODATAKA

## Ciljevi i dizajn baze podataka

Izrada pouzdane i reprezentativne baze podataka jedan je od ključnih koraka u razvoju modela strojnog učenja za klasifikaciju biljaka. Kvaliteta, raznolikost i veličina skupa podataka izravno utječu na točnost i sposobnost generalizacije istreniranog modela. Budući da ne postoji javno dostupan skup podataka koji bi obuhvatio ciljane biljne vrste relevantne za ovo istraživanje, bilo je nužno konstruirati vlastitu bazu posebno prilagođenu zadatku klasifikacije.

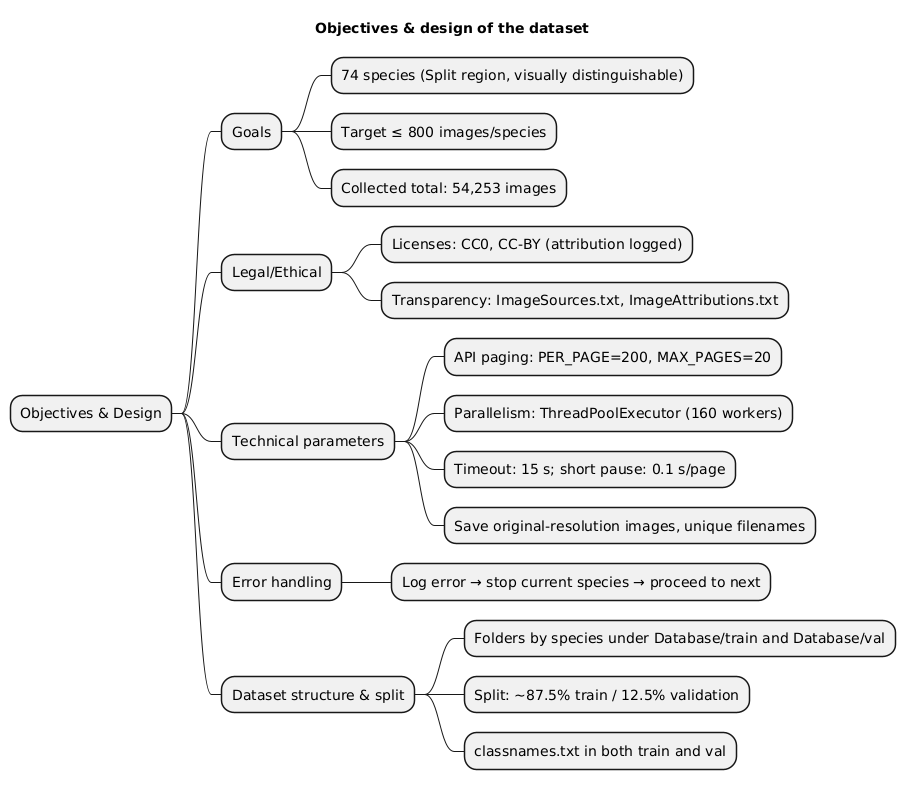
Baza izrađena u sklopu ovog rada obuhvaća 74 vrste biljaka i drveća, većinom onih koje se često nalaze na području Splita i koje su međusobno jasno vizualno prepoznatljive. Kriterij odabira vrsta bio je njihova lokalna rasprostranjenost te praktična upotrebljivost za kasniju automatiziranu identifikaciju. Cilj je bio prikupiti do 800 slika po vrsti, kako bi se obuhvatila dovoljna varijabilnost u osvjetljenju, kutu snimanja, godišnjem dobu i pozadini. Nakon provedbe čitavog postupka prikupljanja i filtriranja, konačna baza sadržava ukupno 54253 slike, što čini dovoljno velik i raznolik skup podataka za treniranje i validaciju modela dubokog učenja.

Fotografije su preuzete putem iNaturalist API-ja, jedne od najvećih globalnih platformi za dokumentiranje bioraznolikosti. Kako bi se osigurala etička i pravna ispravnost, korištene su isključivo slike pod licencama CC0 i CC-BY. Kod fotografija pod licencom CC-BY sve zasluge, ime autora i poveznica na izvor, zabilježene su u posebnu datoteku ImageAttributions.txt, dok je u datoteci ImageSources.txt zabilježen popis svih URL-ova, uz oznaku pripadaju li skupu za treniranje ili skupu za validaciju. Na taj je način zajamčena transparentnost i ponovljivost čitavog procesa.

S tehničke strane, sustav je dizajniran s naglaskom na učinkovitost i otpornost na greške. API se dohvaćao stranica po stranica, s maksimalno 200 rezultata po stranici i do 20 stranica po vrsti. Proces preuzimanja slika bio je paraleliziran korištenjem 160 niti (engl. Threads), što je omogućilo brzo i učinkovito prikupljanje velikog broja slika. Svaka slika spremljena je u izvornoj rezoluciji, pod jedinstvenim imenom u odgovarajući direktorij vrste. Mehanizmi za obradu pogrešaka osigurali su da se eventualni problemi poput nedostupni server ili istek vremena zahtjeva pravilno zabilježe, nakon čega se preuzimanje za tu vrstu prekida i prelazi na sljedeću. Time je spriječeno zadržavanje cijelog procesa na jednoj problematičnoj vrsti.

Prikupljene slike organizirane su u strukturirane direktorije, a baza je kasnije podijeljena na dva skupa, jedan za treniranje i jedan za validaciju. Otprilike 87,5 % slika dodijeljeno je za treniranje, dok je 12,5 % izdvojeno za validaciju. U obje skupine automatski je generirana datoteka classnames.txt s popisom svih 74 vrste, čime se osigurava konzistentnost između baze i modela tijekom faze treniranja.

Ovaj odjeljak definira ciljeve, obuhvat i smjernice dizajna baze podataka. U sljedećim odjeljcima detaljnije će se prikazati izvori podataka, tehnička implementacija sustava za preuzimanje slika, organizaciju i strukturu baze, te konačni rezultati njezine izrade, a na slici Slika 3‑1 vidljiva je umna mapa svega spomenutog u ovom odjeljku.

[](//www.plantuml.com/plantuml/png/LLF1RjD04BtdA-P2LQAqIKf3AoEaXZPKbOg8I0zmGkFrr5vOxvgTsT1UE29nvHluidu9utLenSNoZjzxyzxC-em5exJMLors8x7Y2DwNNyY8tHB37bJ4jlOGRa4QWWe5cMGq6l-FkiYeqNWCbm4TwyiOJew1Ep9M0VkRpbc1IBKDVW9RomcTkuVAibXV9yiDbewUPzuDnfe47dxzWTFP36oBDV5q9vKHvy4vxKqLI11q1IoE9qUB5ncQ8N1DDRhfIXfhq6NEjJNacRY0y_FPH1y7Rpx1Fef4MoPHM-12NLEryn3HSuUHlBaluAhNtOGK3V6XtCbag2oVoBayweba6jyt1MLZIq9nyBnSNscfrhG5h5SVFg-NbwlNHpErycxvCPysUipGjJ8reBFS5d3JHC9g7O9RtP594YBipr_Eu7k8NrLxPzUs598KC5y0lm9kGXJjbfWAc1tEWQVQcZ9qWrk24AqwGNSGYOFB2NR3cq3ozbiYkBMEFBR3E5SnQjy6VUKqm33FK0FbyiFFty0IEZ0fHlBokFN-GnU3eRmey3ICw6AuGyg9oKYA17jASLQoxjlWAeeCvVsZJl9QoRHIYLE9Q3sebwVIThVaVCqA-75wShXu1WDm2lEZ_gGOMs4VDKEDGsODE6mK55W6QU19lDSy8r_z-pV-0W00)

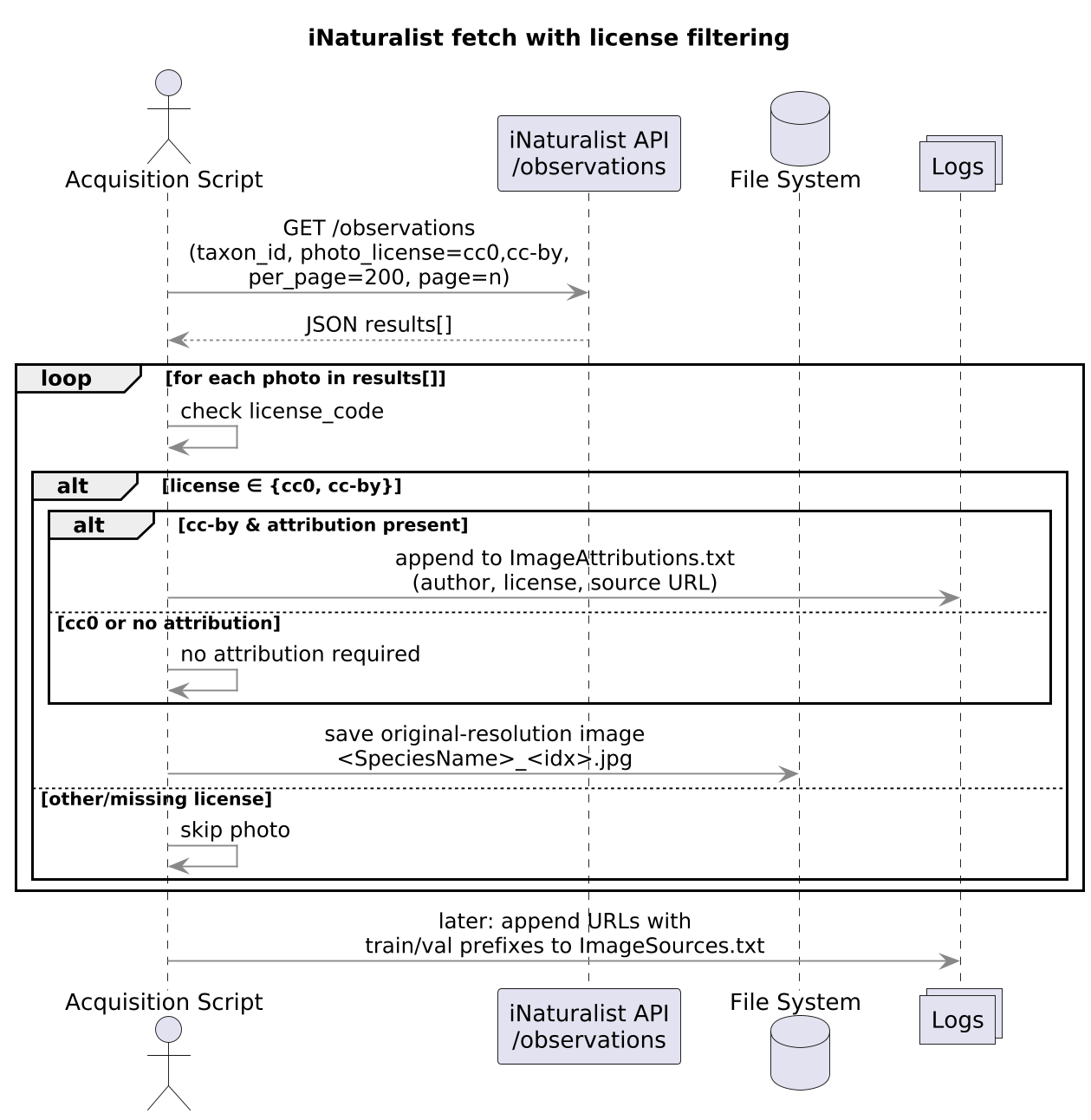
Slika 3‑1 Sažetak ciljeva, pravnih okvira, tehničkih parametara i planirane strukture baze podataka (izrađeno u web editoru <https://plantuml.com/>)

## Izvori podataka

Primarni izvor slika u ovom radu jest platforma iNaturalist, dostupna putem javnog REST API-ja koji omogućuje dohvat opažanja biljnih i životinjskih vrsta zajedno s pripadajućim metapodacima. Odabir iNaturalista motiviran je opsegom i kvalitetom dostupng sadržaja te postojanjem strukturiranih polja koja su ključna za istraživačku upotrebu, uključujući naziv vrste, poveznice na fotografije u više rezolucija, kod licence i podatke o autorima. Takva organizacija podataka omogućuje transparentnu i ponovljivu izgradnju skupa, uz precizno praćenje podrijetla svake pojedinačne slike.

Pristup podacima ostvaren je pozivima na krajnju točku, observations, uz korištenje parametara koji su definirani u sklopu skripte za akviziciju. Za svaku vrstu iz unaprijed pripremljenog popisa, pohranjenog u datoteci SpeciesList.py, dohvaćaju se opažanja filtrirana po taksonomskom identifikatoru i ograničena na fotografije s prihvatljivim licencama. Paginacija je namještena tako da se po stranici preuzima do dvjesto zapisa, dok je ukupan broj stranica po vrsti ograničen na dvadeset, čime se postiže razumna gornja granica preuzetih slika i kontrolira opterećenje udaljene usluge. Kako bi se smanjio rizik od prekoračenja ograničenja poslužitelja i ublažile kratkotrajne nestabilnosti mreže, između uzastopnih zahtjeva uvodi se vrlo kratka odgoda, a svaki zahtjev ima podešen vremenski prag nakon kojega se smatra neuspjelim. U slučaju pogreške zapisuje se opis problema i dohvat se za tu vrstu prekida, pri čemu proces odmah nastavlja s idućom vrstom, što omogućuje robusno i učinkovito prikupljanje bez zastoja.

Korištene su isključivo fotografije pod licencama CC0 i CC-BY. Licenca CC0 podrazumijeva odricanje nositelja prava i stavljanje djela u javno vlasništvo, što omogućuje slobodno korištenje i prerade bez dodatnih ograničenja. Licenca CC-BY dopušta široku upotrebu, ali uz obvezno navođenje autorstva, iz tog je razloga svaka slika s ovom licencom, kada su podaci dostupni, popraćena atribucijom koja uključuje ime autora i izvorni URL. Ove su informacije sustavno zapisane u datoteku ImageAttributions.txt, dok se potpuni popis svih URL-ova koristi u datoteci ImageSources.txt, u kojoj je za svaki zapis naznačeno pripada li trenirajućem ili validacijskom skupu. Izbor upravo ovih licenci postavlja jasne pravne okvire i uklanja neizvjesnosti oko daljnje znanstvene i obrazovne upotrebe, restriktivnije licence koje uvjetuju, primjerice, nekomercijalnu upotrebu ili zabranu prerada nisu obuhvaćene, kako bi se osigurala potpuna kompatibilnost s postupkom treniranja, evaluacije i kasnijeg prikaza rezultata. Na slici Slika 3‑2 se vidi kako skripta dohvaća i filtrira slike po licencama.



Slika 3‑2 Tijek dohvaćanja i filtriranja po licenci (izrađeno u web editoru <https://plantuml.com/>)

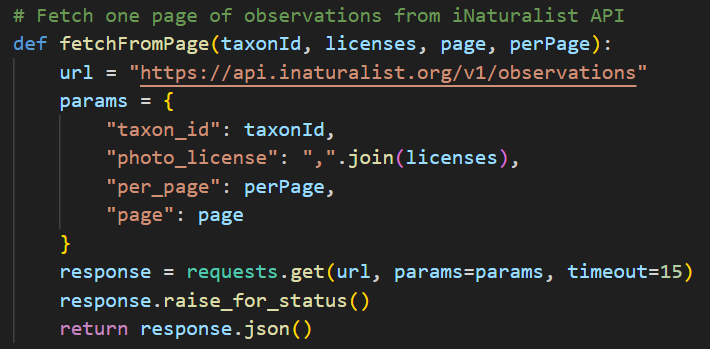
S obzirom na kvalitetu ulaznih podataka, slike se preuzimaju u izvornoj rezoluciji kada je to moguće, pri čemu se na temelju metapodataka formira izravna poveznica na izvornu varijantu fotografije. Svaka se datoteka pohranjuje pod jednoznačnim nazivom koji uključuje ime vrste i indeks, u pripadajući direktorij te vrste, čime se osigurava konzistentnost i lako pretraživanje. Na kraju postupka prikupljanja, cjelokupan popis izvora i pripadne atribucije omogućuju potpunu sljedivost skupa te olakšavaju naknadnu rekonstrukciju baze na temelju istih URL-ova, što je detaljnije opisano u posebnom pododjeljku o rekonstrukciji.

Opisani pristup spaja tehničku učinkovitost s pravnom i metodološkom urednošću. iNaturalist API pruža standardiziran i bogat skup metapodataka, čime se omogućuje dosljedno filtriranje po licenci i vrsti, kontrolirano preuzimanje velikog broja slika uz paginaciju i pauze između zahtjeva, te sustavno bilježenje izvora i atribucija. Rezultat je pouzdan i transparentan temelj na kojem se u sljedećim pododjeljcima gradi organizacija direktorija, nasumična podjela na trenirajući i validacijski dio te cjelokupna reproducibilnost izrade baze.

## Tehnička izvedba prikupljanja slika

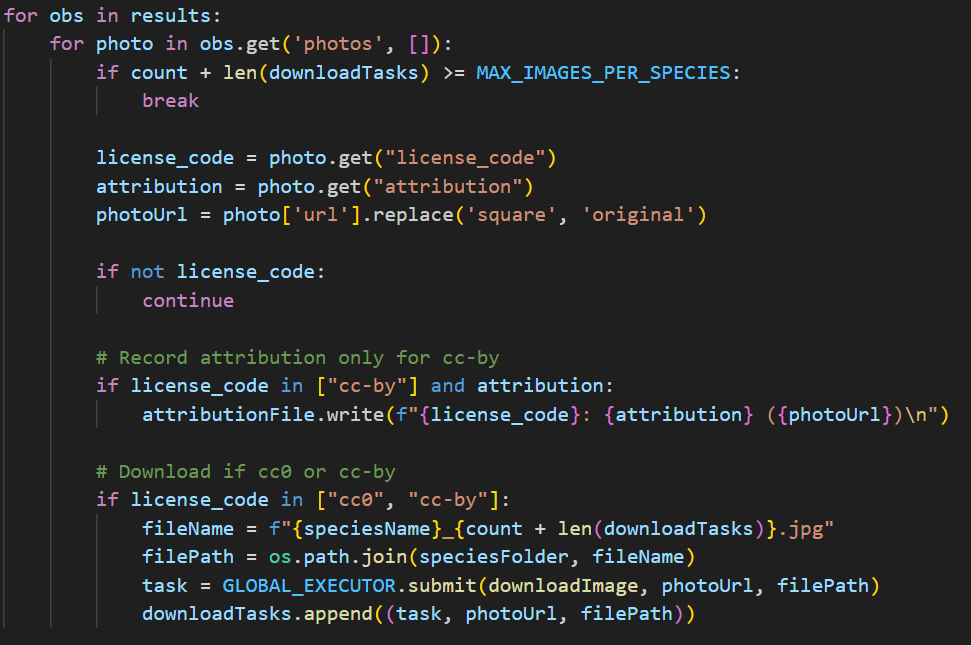
Postupak akvizicije provodi se skriptom u programskom jeziku Python koja nad iNaturalist API-jem izvršava seriju paginiranih upita te asinkrono preuzima fotografije prema unaprijed zadanim ograničenjima. Ulaz u proces jest popis ciljnih vrsta zapisan s njihovim taksonomskim identifikatorima unutar SpeciesList.py datoteke, dok su izlaz strukturirani direktoriji s datotekama slika i popratnom dokumentacijom izvora i atribucija. Fokus implementacije je na učinkovitosti preuzimanja i reproducibilnosti, uz jasna pravila filtriranja licenci i determinističko imenovanje datoteka.

Komunikacija s API-jem ostvaruje se putem krajnje točke observations. Za svaku vrstu generira se upit s parametrima taxon\_id, photo\_license, per\_page i page, pri čemu photo\_license eksplicitno ograničava rezultate na CC0 i CC-BY. Vrijednosti paginacije odabrane su tako da se preuzme do 200 zapisa po stranici, uz najviše 20 stranica po vrsti. Svaki zahtjev ima vremensko ograničenje od 15 sekundi, a između uzastopnih stranica uvodi se kratka odgoda kako bi se smanjilo opterećenje udaljene usluge i izbjegla neželjena ograničenja na strani poslužitelja. Slika ispod pokazuje način formiranja i provjere odgovora.



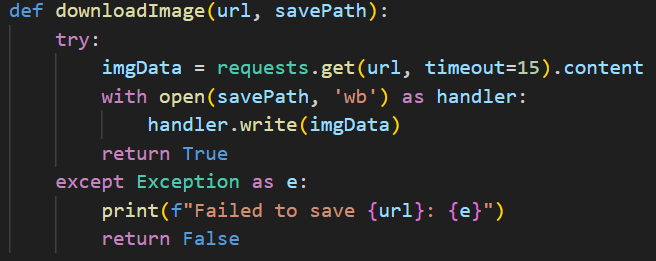
Slika 3‑3 Formiranja i provjere odgovora

Akvizicija je organizirana po vrstama, za svaku vrstu otvara se ciljna mapa na putanji Database/train/<NazivVrste> i pokreće iteracija kroz stranice i opažanja sve dok se ne dostigne gornja granica od najviše 800 slika po vrsti, ne iscrpe rezultati ili ne nastupi greška. Iz svakog opažanja čitaju se metapodaci fotografija. Ako fotografija nema odgovarajući kod licence, preskače se, u suprotnom se priprema URL izvorne rezolucije zamjenom segmenta square s original, čime se izbjegavaju umanjene inačice. Fotografije s licencom CC-BY dodatno prate atribucije autora, koje se zapisuju u Sources/ImageAttributions.txt zajedno s URL-om kako bi kasnije citiranje bilo jednoznačno i provjerljivo. Na slici Slika 3‑4 prikazan je sažetak logike obrade jednog opažanja.



Slika 3‑4 Sažetak programske logike za jedno opažanje

S ciljem ubrzanja procesa koristi se globalni spremnik niti ,ThreadPoolExecutor, s najviše 160 paralelnih zadataka kako bi se u potpunosti iskoristila brzina mreže od 1gigabita po sekundi. Preuzimanje je granulirano na razinu datoteke i svaka se slika zapisuje pod jedinstvenim imenom oblika <NazivVrste>\_<indeks>.jpg. Ovako formirani nazivi omogućuju determinističko adresiranje primjera te olakšavaju kasnije rekonstruiranje i provjeru. Nakon što se batch zadataka preuzimanja preda izvršitelju, glavna nit čeka dovršetak tih zadataka prije prelaska na sljedeću stranicu ili zaključivanja vrste, čime se postiže kontrolirano korištenje mrežnih i diskovnih resursa. Funkcija za preuzimanje slika koja provodi osnovno rukovanje pogreškama i vraća uspješnost vidljiva je na slici Slika 3‑5.



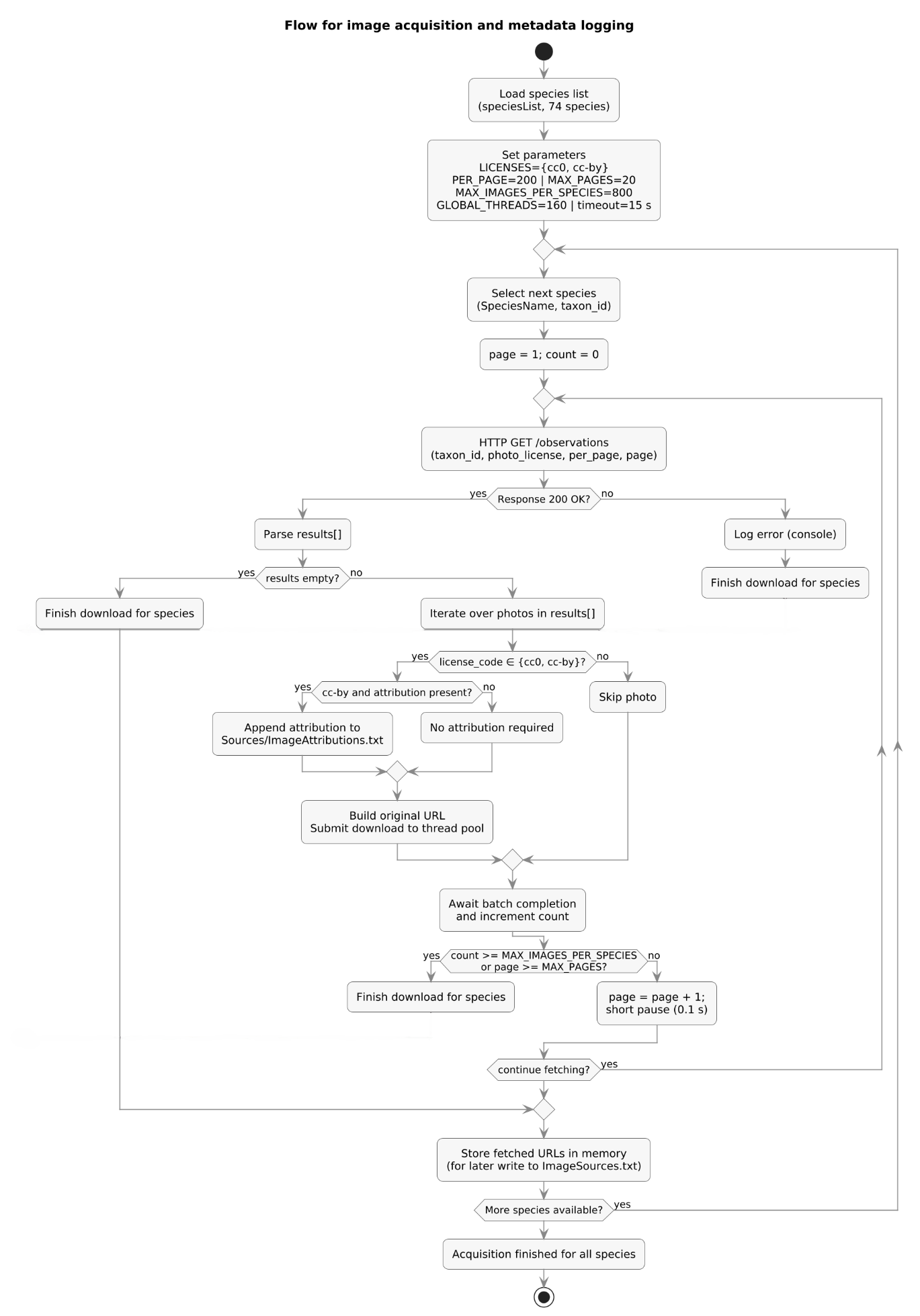
Slika 3‑5 Funkcija za preuzimanje slika

Rukovanje pogreškama oslanja se na uzorak „zabilježi i nastavi s idućom jedinicom posla“. U slučaju iznimke pri dohvaćanju stranice, skripta zapisuje opis greške i prekida akviziciju za tu vrstu, odmah prelazeći na sljedeću. Takav kompromis izbjegava zaglavljenja i nepotrebne ponovne pokušaje u uvjetima povećane nestabilnosti. Istodobno, budući da se u Sources/ImageSources.txt naknadno zapisuje točan popis URL-ova po vrsti, omogućeno je kasnije ciljano ponavljanje neuspješnih preuzimanja u zasebnoj fazi, kao i potpuna rekonstrukcija baze na temelju logova. Završetkom obrade jedne vrste, sve uspješno preuzete slike ostaju u njezinoj mapi unutar train, a metapodaci su spremni za sljedeći korak nasumične podjele na trenirajući i validacijski skup, o čemu je detaljnije riječ u odjeljku 3.4.

Posebna pažnja posvećena je transparentnosti i ponovljivosti. Svaki URL koji je doista korišten zapisuje se s oznakom pripada li trenirajućem ili validacijskom skupu, a atribucije za CC-BY bilježe se odvojeno. Time se postiže sljedivost od modela natrag do izvora slike i autora, što je važno i u akademskom i u produkcijskom okruženju. Imenovanje datoteka i struktura direktorija pojednostavljuju ručne inspekcije i skriptirano pretraživanje, a ograničenja poput maksimalnog broja slika po vrsti i maksimalnog broja stranica štite od neželjenog rasta baze ili pretjeranog opterećenja udaljene usluge.

U pogledu optimizacija, trenutna verzija akvizicije koristi requests.get(...).content za preuzimanje, što je praktično i dovoljno robusno za prosječne veličine slika; alternativno strujanje (stream=True) i zapis u komadima, kakvo se primjenjuje u skripti za rekonstrukciju, može se uvesti i ovdje kada se očekuju iznadprosječno velike datoteke ili sporiji diskovni sustavi. Dodatna poboljšanja mogla bi uključivati heuristike za deduplikaciju na temelju sažetaka, filtar minimalne rezolucije prije uvrštavanja u skup te ograničenja po hostu radi ravnomjernije potrošnje mrežnih resursa; takve su nadogradnje kompatibilne s postojećom arhitekturom i mogu se dodati bez promjene protokola s API-jem.

Na kraju ovog koraka, baza podataka postoji u „prolaznom“ obliku u kojem su sve slike pohranjene unutar Database/train/<NazivVrste>, a logovi izvora i atribucija su aktualni. Sljedeći odjeljak obrađuje formalnu podjelu na trenirajući i validacijski skup, fizičko premještanje odabranih primjera u Database/val/<NazivVrste> i generiranje pomoćnih datoteka za rad s modelom. Dijagram toka skripte može se vidjeti na slici Slika 3‑6.

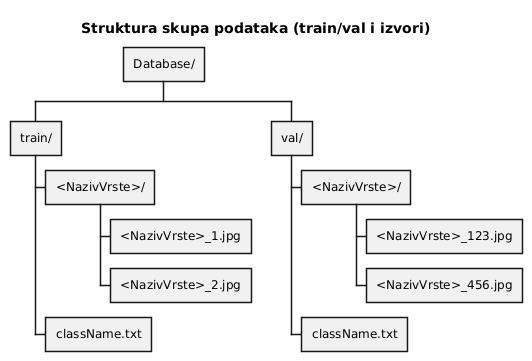


Slika 3‑6 Dijagram toka za izradu baze podataka (izrađeno u web editoru <https://plantuml.com/>)

## Organizacija i podjela podataka

Organizacija skupa podataka izvedena je tako da neposredno podržava tipičan tijek treniranja modela računalnog vida i da ostane maksimalno transparentna i ponovljiva. Svi uzorci razvrstani su po klasama na razini vrste te fizički razdvojeni u dio za treniranje i dio za validaciju. Time se izbjegava nenamjerno „curenje“ podataka između faza treniranja i provjere, a istodobno se zadržava jasna veza između svake datoteke i pripadne klase.

Struktura direktorija slijedi jednostavan, ali u praksi provjeren raspored. Za svaku je vrstu izrađen poddirektorij unutar mape Database/train, a nakon podjele i odgovarajući poddirektorij unutar Database/val. Datoteke slika imenuju se deterministički obrascem <NazivVrste>\_<indeks>.jpg, što olakšava ručnu inspekciju i skriptirano pretraživanje te sprječava kolizije imena. U korijenu i trenirajućeg i validacijskog dijela automatski se generira datoteka classnames.txt s popisom svih sedamdeset i četiri klase. Ova lista služi kao jedini izvor istine za mapiranje redoslijeda klasa na indekse etiketa tijekom treniranja i evaluacije te omogućuje potpunu dosljednost između skupa podataka i modela. Struktura direktorija prikazana je na slici Slika 3‑7.



Slika 3‑7 Struktura skupa podataka (izrađeno u web editoru <https://plantuml.com/>)

Podjela na trenirajući i validacijski dio provedena je nasumično, zasebno za svaku vrstu, kako bi se očuvala proporcionalnost klasa i izbjegla sistemska pristranost. Odabrani omjer iznosi približno 87,5 % za treniranje i 12,5 % za validaciju, pri čemu se vodi računa da svaka klasa ima barem jedan primjer u validacijskom skupu, čak i kada je broj dostupnih slika malen. Implementacijski, odabir validacijskih uzoraka radi se nad popisom već preuzetih datoteka za pojedinu vrstu, a zatim se odabrane datoteke fizički premještaju iz trenirajuće mape u odgovarajuću validacijsku mapu. Takva fizička separacija onemogućuje slučajno korištenje istih datoteka u obje faze te pojednostavljuje konfiguraciju alata za treniranje koji očekuju konvencionalan raspored mapa po klasi. Funkcija za podjelu na dio za treniranje i dio za validaciju vidljiv je na slici Slika 3‑8.



Slika 3‑8 Funkcija za raspodjelu slika u na set za treniranje i set za validaciju

Kako bi se očuvala sljedivost i omogućila rekonstrukcija, sve iskorištene poveznice na izvorne datoteke bilježe se u Sources/ImageSources.txt uz prefiks koji naznačuje pripadnost trenirajućem ili validacijskom dijelu. Ovaj manifest nastaje nakon podjele i predstavlja točan presjek skupa podataka u trenutku treniranja. U zasebnoj datoteci Sources/ImageAttributions.txt evidentiraju se i atribucije za fotografije pod licencom CC-BY, čime se zadovoljavaju pravni i etički zahtjevi te omogućuje naknadna revizija izvora. U slučaju ponovne izgradnje skupa na drugom računalu ili nakon gubitka podataka, ovi zapisi služe kao jedini potrebni ulaz, što je u potpunosti opisano u poglavlju o rekonstrukciji.

Posebna je pozornost posvećena integritetu i izbjegavanju curenja podataka. Budući da se validacijski uzorci biraju iz već formiranog skupa datoteka za pojedinu vrstu, ista se slika ne može pojaviti u obje particije. Ipak, kada isti izvor sadrži više vizualno vrlo sličnih kadrova, preporučljivo je u budućim iteracijama uvesti dodatne provjere, primjerice deduplikaciju na temelju sažetaka (hash) ili prag minimalne rezolucije, kako bi se otkrile i isključile degradirane ili duplicirane datoteke. Isto tako, iako trenutna podjela ne provodi eksplicitno balansiranje klasa, neravnoteža se u praksi može ublažiti tehnikama obogaćivanja podataka u fazi treniranja ili ponderiranjem gubitka po klasi, svjesno je odlučeno da se balansiranje ne uvodi u fazi organizacije skupa podataka kako bi se zadržala vjernost realnoj raspodjeli dostupnih primjera.

U implementaciji se vodi računa o idempotentnosti i otpornosti na prekide. Kreiranje odredišnih direktorija tolerantno je na postojeće mape, a premještanje datoteka obavlja se atomarno gdje god je to moguće. U slučaju prekida procesa, ponovno pokretanje neće duplicirati već premještene datoteke niti će promijeniti već donesene odluke o pripadnosti trenirajućem ili validacijskom dijelu, budući da se one odražavaju u manifestu URL-ova. Ovakav pristup pojednostavljuje verzioniranje skupa i omogućuje usporedivost eksperimenata, jer različite konfiguracije treniranja mogu referencirati isti, stabilizirani presjek podataka.

Opisani raspored direktorija, determinističko imenovanje datoteka, eksplicitna evidencija izvora i nasumična, po-klasna podjela s minimalnim brojem validacijskih primjera čine cjelinu koja je odmah upotrebljiva u standardnim trenirnim okruženjima i lako prenosiva između računala. U sljedećem poglavlju o treniranju modela ova će struktura biti izravno korištena za izgradnju i procjenu klasifikatora.

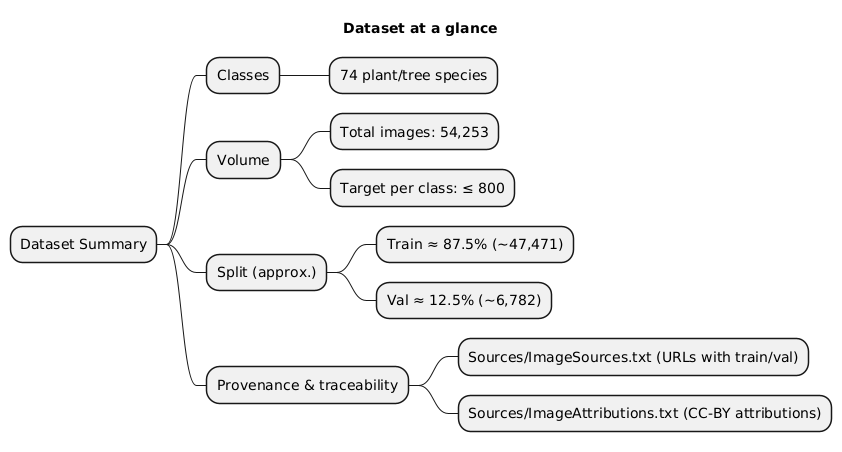
## Rezultati izrade baze

Provedenim postupkom akvizicije i organizacije oblikovana je baza podataka prilagođena zadatku klasifikacije biljaka i drveća u lokalnom kontekstu. Baza obuhvaća ukupno 74 vrste, za koje je prikupljeno 54 253 fotografija. Ciljani prag od najviše 800 slika po vrsti nije uvijek dosegnut zbog ograničene dostupnosti ili licencnih uvjeta pojedinih zapisa, no ostvareni ukupni obujam i raznolikost vizualnih uvjeta, različita osvjetljenja, pozadine, kutovi snimanja i godišnja doba, čine skup dostatno bogatim za treniranje i procjenu modernih konvolucijskih arhitektura.

Podaci su podijeljeni na trenirajući i validacijski dio sukladno omjeru određenom u skripti. U trenirajući skup raspoređeno je približno 47 471 slika (oko 87,5 %), a u validacijski 6 782 slike (oko 12,5 %). Promatrano prosječno po klasi, to odgovara približno 642 primjera po vrsti u trenirajućem skupu i 92 primjera po vrsti u validacijskom skupu, uz napomenu da je raspodjela po klasama neujednačena jer odražava realnu dostupnost izvornog materijala. Nasumična podjela provedena je zasebno za svaku vrstu, pri čemu je osigurano da svaka klasa ima barem jedan primjer u validacijskom skupu te da ne dolazi do preklapanja istih datoteka između particija.

Cjelokupna izrada baze popraćena je pomoćnim artefaktima koji osiguravaju sljedivost i ponovljivost. Datoteka Sources/ImageSources.txt sadrži popis svih URL-ova korištenih slika s oznakom pripadnosti trenirajućem ili validacijskom skupu, dok Sources/ImageAttributions.txt evidentira potrebne atribucije za zapise pod licencom CC-BY. Ovakav manifest omogućuje reviziju izvora i, prema potrebi, potpunu rekonstrukciju baze na drugom sustavu bez promjene sastava podataka. Ujedno, determinističko imenovanje datoteka i dosljedna struktura direktorija olakšavaju ručnu inspekciju i automatizirano korištenje u treniranju. Pregled baze podataka može se vidjeti na slici Slika 3‑9.

Ako su tijekom akvizicije prisutne uobičajene tehničke teškoće kao što su povremeni prekidi veze ili nedostupni resursi, mehanizmi za bilježenje pogrešaka i tolerantno ponašanje skripte sprječavaju zastoj postupka i minimaliziraju gubitak podataka. Potencijalna buduća poboljšanja obuhvaćaju deduplikaciju na razini sažetaka, pragove minimalne rezolucije i jednostruko balansiranje klasa prije treniranja; ta su unaprjeđenja kompatibilna sa sadašnjom arhitekturom i ne mijenjaju zaključak da je izrađena baza dovoljno velika, raznolika i dobro dokumentirana za daljnju eksperimentalnu evaluaciju modela u sljedećem poglavlju.

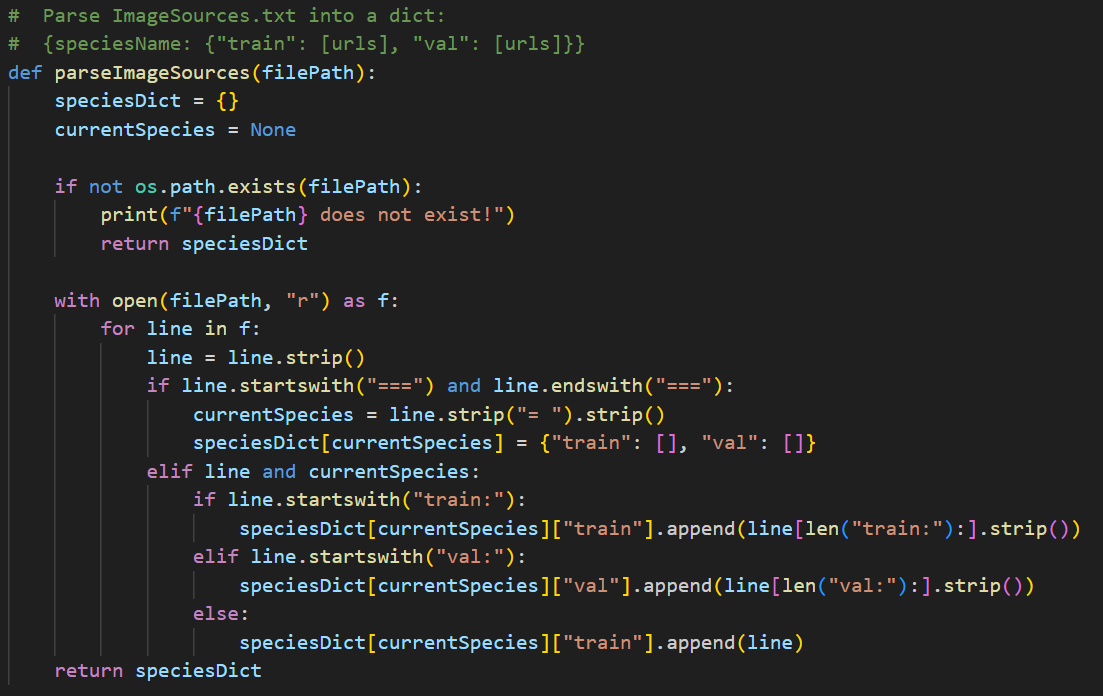


Slika 3‑9 Pregled baze podataka (izrađeno u web editoru <https://plantuml.com/>)

## Rekonstrukcija baze podataka

Rekonstrukcija baze podataka osigurava ponovljivost, prenosivost i dugoročnu održivost rezultata ovog rada. Polazišna pretpostavka jest da su tijekom akvizicije i organizacije podataka, kako je opisano u odjeljcima 3.3 i 3.4, generirani manifesti izvora i atribucija, ImageSources.txt i ImageAttributions.txt. Na temelju tih zapisa, bez ponovnog pozivanja iNaturalist API-ja, moguće je na novom sustavu ili nakon gubitka podataka ponovno izgraditi doslovno isti raspored direktorija uz isti presjek skupova, čime se jamči da će kasniji eksperimenti nad rekonstruiranom bazom dati usporedive rezultate.

Skripta za rekonstrukciju polazi od datoteke ImageSources.txt, koja je strukturirana po vrstama i particijama. Naslove odjeljaka označene linijama u obliku „=== NazivVrste ===“ skripta interpretira kao početak bloka za pojedinu vrstu, dok se u nastavku nalaze redci s prefiksima „train:“ i „val:“ koji sadrže izvorne URL-ove slika dodijeljenih trenirajućem i validacijskom skupu. Parsiranjem ove datoteke gradi se u memoriji preslikavanje vrste na dva popisa poveznica, čime se očuva ista podjela koja je korištena pri treniranju modela. Budući da je ImageAttributions.txt već nastao u fazi akvizicije, njegova se uloga u rekonstrukciji svodi na dokumentiranje pravnih i etičkih metapodataka. Rekonstrukcija ne dohvaća atribucije iz izvora nego samo ponovno preuzima slikovne sadržaje. Funkcija za parsiranje datoteke s URL-ovima prikazana je na slici Slika 3‑10.

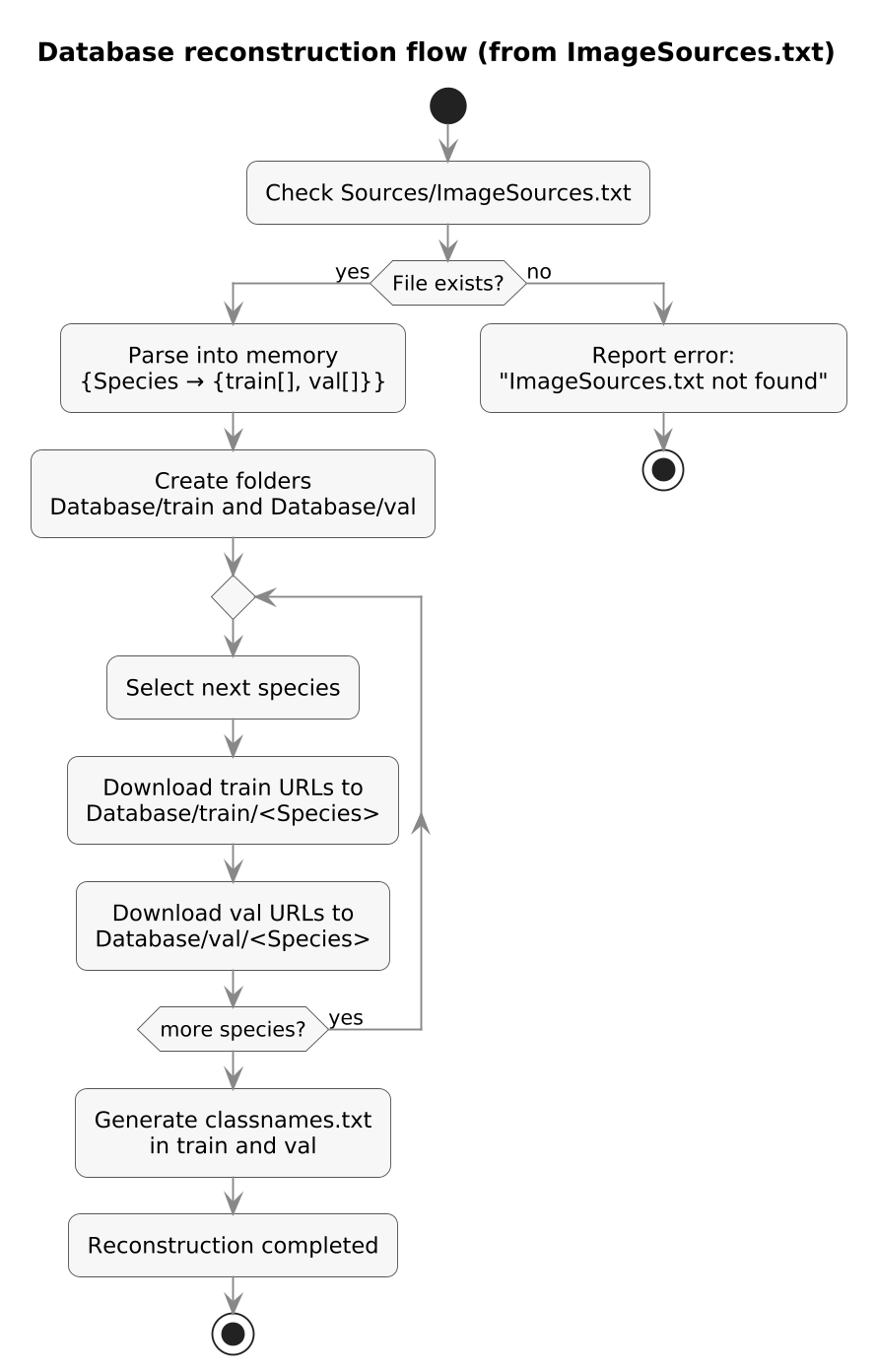


Slika 3‑10 Funkcija za parsiranje datoteke sa izvorima slika

Preuzimanje se provodi paralelno, uz korištenje spremnika niti s visokim stupnjem usporednosti, kako bi se proces značajno ubrzao i učinkovitije iskoristili mrežni i diskovni resursi. Za razliku od inicijalne akvizicije, gdje su slike preuzimane preko API-ja, ovdje se radi o izravnom dohvaćanju već poznatih URL-ova, uz strujni zapis na disk i fragmentiranje u manje blokove podataka. Takav način zapisa smanjuje potrošnju memorije i povećava robusnost u slučaju sporijih veza ili većih datoteka. Svaka je slika deterministički imenovana prema uzorku <NazivVrste>\_<indeks>.jpg i smješta se u odgovarajući direktorij ovisno o pripadnosti trenirajućem ili validacijskom skupu, čime se dosljedno reproducira struktura Database/train/<NazivVrste> i Database/val/<NazivVrste>.

Skripta je osmišljena tako da bude idempotentna i nastaviva. Ako datoteka s očekivanim nazivom već postoji, preskače se ponovni download, što omogućuje sigurno ponavljanje procesa nakon prekida ili djelomične rekonstrukcije. Neuspjeli pokušaji preuzimanja evidentiraju se u standardni izlaz s opisom uzroka, poput isteknuće vremena, nedostupan resurs, pogreška poslužitelja, no takvi pojedinačni padovi ne zaustavljaju cjelokupan proces. Rekonstrukcija za tu sliku se preskače, a izvođenje se nastavlja za preostale URL-ove i vrste. Na kraju procesa automatski se ponovno generiraju datoteke classnames.txt u korijenu train i val direktorija na temelju popisa vrsta iz SpeciesList.py, čime se osigurava usklađenost između rekonstruirane baze i trening skripti koje očekuju te pomoćne datoteke Dijagram toka skripte za rekonstrukciju vidljiv je na slici Slika 3‑11.

Ograničenja ove metode proizlaze iz ovisnosti o vanjskim izvorima: s vremenom pojedini URL-ovi mogu zastarjeti ili postati nedostupni. To ne narušava strukturu niti integritet cjelokupnog skupa, ali može smanjiti broj dostupnih primjera za pojedine vrste. Kao razumna nadogradnja preporučuje se uvesti strategiju ponovnih pokušaja s postupnim čekanjem, provjeru tipa sadržaja i minimalne rezolucije prije spremanja te izračun kriptografskih sažetaka radi detekcije dupliciranih ili oštećenih datoteka. Unatoč tim potencijalnim poboljšanjima, opisani postupak već ispunjava osnovni cilj, pouzdano i doslovno ponovno izgraditi skup podataka na temelju manifest-datoteka, bez ovisnosti o trenutačnom stanju API-ja ili promjenama na izvoru, te time omogućiti potpunu ponovljivost treninga i evaluacije u sljedećim poglavljima.



Slika 3‑11 Dijagram toka za rekonstrukciju baze podataka (izrađeno u web editoru <https://plantuml.com/>)

# TRENIRANJE I EVALUACIJA MODELA

## Ciljevi treniranja i metodologija

Cilj ovoga poglavlja jest izgraditi i procijeniti model za automatsku klasifikaciju 74 biljnih vrsta na skupu opisanom u poglavlju 3, koji ukupno sadrži 54 253 slika. Primarni kriterij uspjeha je točnost na validacijskom skupu, uz stabilne krivulje učenja bez znakova pretreniranosti. Dodatni, praktični cilj jest da model bude upotrebljiv na mobilnom uređaju, u ovom slučaju Android, pa se tijekom dizajna vodi računa o omjeru točnosti i računalne učinkovitosti.

Metodologija se temelji na nadziranom učenju s već pripremljenom podjelom na train i val unutar strukture baze podataka. Ulazne slike prolaze kroz standardiziran pipeline, u treniranju se primjenjuju prostorne i fotometrijske transformacije koje uvode realistične varijacije, slučajni izrez do ciljnih 300×300 piksela, horizontalna i vertikalna zrcaljenja, rotacije do 30°, blage afine i perspektivne transformacije, prilagodbe svjetline, kontrasta, zasićenja, nijanse te RandomErasing, dok je validacijski tok determinističan, promjena veličine, središnji izrez na 300×300, normalizacija. Sve slike normalizirane su na ImageNet statistiku, u skladu s inicijalizacijom modela.

Arhitektura klasifikatora je EfficientNet-B3 inicijalizirana unaprijed naučenim težinama, a završni klasifikacijski sloj zamijenjen je slojem čija dimenzija odgovara broju klasa, 74. Treniranje se provodi križno-entropijskim gubitkom i optimizatorom AdamW s korakom učenja 1×10⁻⁴. Maksimalno trajanje iznosi 30 epoha, uz rano zaustavljanje s odgodom od 4 epohe, tj. ako se validacijska točnost ne poboljša kroz četiri uzastopne epohe, proces se prekida. Tijekom svake epohe računa se gubitak u treniranju i validaciji te točnost na validacijskom skupu, zadržavaju se najbolje težine prema validacijskoj točnosti, a po završetku model se vraća na taj optimum. Trening se izvršava na NVIDIA RTX 4060 (laptop) grafičkoj kartici uz veličinu paketa 8 i 4 radnika za učitavanje podataka, što je pragmatičan kompromis s obzirom na memorijska ograničenja.

Evaluacija u ovom koraku usredotočena je na vrijednosti gubitka i točnosti po epohi te pripadajuće artefakte koji se automatski generiraju, graf krivulja TrainingPlot.png i zapis točnosti po epohi , ValAccuracyPerEpoch.txt. Ovi rezultati služe za kvalitativnu provjeru stabilnosti učenja i usporedbu različitih pokretanja. Najbolji model potom se serijalizira i priprema za izvoz u mobilno-prikladan format, što je tema završnih odjeljaka ovoga poglavlja.

## Priprema podataka

Skup podataka učitava se iz direktorijske strukture opisane u poglavlju 3, pri čemu Database/train/<NazivVrste> i Database/val/<NazivVrste> definiraju klase. Učitavanje je realizirano preko torchvision.datasets.ImageFolder, a nazivi klasa dobivaju se iz trainingDataset.classes i dalje koriste za oblikovanje klasifikacijske glave i generiranje labels.txt za aplikaciju.

Tijekom treniranja primjenjuje se skup transformacija koje istodobno standardiziraju ulaz i obogaćuju podatke realističnim varijacijama. Slike se stohastički kadriraju na 300×300 piksela preko RandomResizedCrop(300, scale=(0.8, 1.0)), zatim slijede horizontalno i vertikalno zrcaljenje te rotacije do 30°. Fotometrijske promjene uvodi ColorJitter, svjetlina i kontrast do 0.4, zasićenje 0.3, nijansa 0.1, a geometrijske varijacije RandomAffine, translacija do 15 %, skala 0.9–1.1 i RandomPerspective, faktor 0.2, vjerojatnost 0.5. Nakon pretvorbe u tenzor (ToTensor) provodi se normalizacija na ImageNet statistiku, mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]), a RandomErasing(p=0.3, scale=(0.02, 0.15)) dodatno simulira parcijalne okluzije i smanjuje prenaučenost na lokalne uzorke.

## Arhitektura modela

Klasifikator je zasnovan na EfficientNet-B3 iz torchvision-a. Inicijalizacija se provodi funkcijom initializeModel(numClasses, device), koja učitava unaprijed istrenirane težine, EfficientNet\_B3\_Weights.DEFAULT, te zamjenjuje završni linearni sloj u model.classifier[1] novim slojem dimenzije numClasses. Time se broj izlaznih logita usklađuje s brojem klasa (74) detektiranim iz classnames.txt. Ostatak mreže, konvolucijski backbone i optimizacije EfficientNeta, ostaje neizmijenjen kako bi se iskoristile generičke vizualne reprezentacije naučene na velikom skupu, npr. ImageNet-u. Model se nakon izmjene premješta na odabrani uređaj, CPU ili GPU, i spreman je za treniranje prema protokolu iz sljedećeg odjeljka. Ulazna rezolucija od 300×300 iz DataUtility usklađena je s kapacitetom B3 varijante i omogućuje dobar omjer točnosti i računalne učinkovitosti za kasniju mobilnu primjenu.

U odabiru EfficientNet-B3 vodilo se računa o ravnoteži između točnosti i računalne učinkovitosti u mobilnom okruženju. Arhitektura koristi ujednačeno skaliranje dubine, širine i rezolucije, što omogućuje da se na ulazu dimenzije 300×300 postigne dobar omjer kapaciteta i troška izvođenja. Prethodno naučene reprezentacije, pretrained težine, služe kao opći vizualni izvor znanja, dok se završna klasifikacijska glava u potpunosti prilagođava ciljnom problemu zamjenom linearnog sloja s 74 izlaza. Time se zadržava snaga backbone-a za ekstrakciju značajki, a prediktor postaje specifičan za biljne vrste u bazi.

## Strategije učenja i postupak treniranja

Strategija učenja u ovom radu temelji se na end-to-end finom podešavanju unaprijed istreniranog EfficientNet-B3 modela na ciljnom skupu od 74 klase. Polazišna točka su težine naučene na općem domenskom skupu , dok se tijekom treniranja ažuriraju svi parametri mreže, uključujući novododani završni klasifikacijsku sloj. Takav pristup omogućuje da se postojeće generičke vizualne reprezentacije adaptiraju na specifičnosti biljnih vrsta, a da se pritom zadrži brzina konvergencije karakteristična za transfer učenje.

Ulazni podaci dolaze iz dva tokova, trenirajućeg i validacijskog, konstruiranih u DataUtility. U trenirajućem toku primjenjuju se prostorne i fotometrijske transformacije, nasumični izrez na 300×300, zrcaljenja, rotacije, perspektivne deformacije, promjene svjetline, kontrasta, zasićenja, nijanse te RandomErasing, dok je validacijski tok determinističan, promjena veličine, središnji izrez na 300×300, normalizacija. U obje faze slike se normaliziraju na ImageNet statistiku, što je usklađeno s inicijalizacijom EfficientNeta.

Funkcionalno, treniranje je organizirano po epohama, a svaka epoha ima dvije faze, train i val. U fazi treniranja model je u načinu train(), izračunava se križno-entropijski gubitak nad izlaznim podatcima, provodi se povratno širenje pogreške i ažuriranje težina optimizatorom AdamW s korakom učenja 1×10⁻⁴. U validacijskoj fazi model je u načinu eval(), gradijenti su isključeni, a mjeri se gubitak i točnost na zasebnom skupu. Veličina paketa postavljena je na 8, a broj radnika za učitavanje na 4, što je učinjeno na takav način da se maksimalno iskoristi hardver, s obzirom na korištenje NVIDIA RTX 4060 (laptop) grafičke kartice i ciljani kompromis između propusnosti i potrošnje memorije.

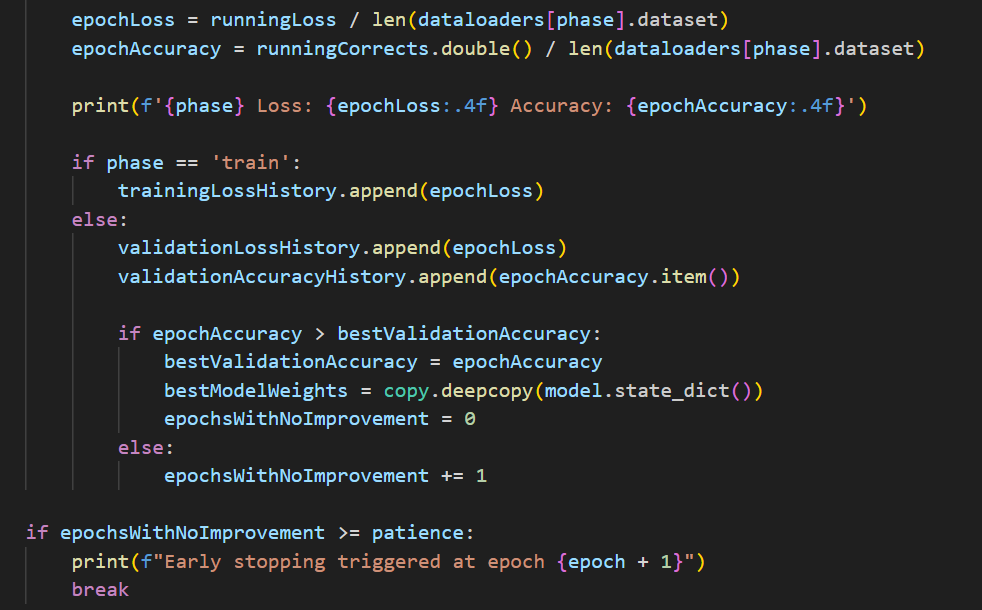
Tijekom svake epohe akumulira se prosječni gubitak po uzorku te broj točnih predikcija iz kojih se računa točnost epohe. U isto vrijeme vodi se povijest metrika, nizovi trenirajućeg i validacijskog gubitka te validacijske točnosti po epohama. Ključan element strategije je rano zaustavljanje s odgodom od četiri epohe, tj. ako se validacijska točnost ne poboljša kroz četiri uzastopne epohe, proces se prekida kako bi se izbjeglo pretreniranje i nepotrebno trošenje računalnih resursa. Paralelno se čuva najbolja instanca modela kada god validacijska točnost ostvari novi maksimum, pohranjuje se duboka kopija trenutnih težina. Po završetku treniranja model se vraća upravo na taj skup težina i kao takav serijalizira za daljnju uporabu.

Opisani postupak rezultira skupom artefakata za praćenje učenja: graf TrainingPlot.png , krivulje trenirajućeg/validacijskog gubitka i validacijske točnosti po epohama, te tekstualni zapis ValAccuracyPerEpoch.txt. Ti se zapisi koriste u idućem odjeljku za formalno izvještavanje i kvalitativnu interpretaciju tijeka učenja, a najbolja pohranjena verzija modela predstavlja polazište za izvoz u mobilni format.

## Protokol evaluacije i izvještavanje

Evaluacija se provodi na validacijskom skupu definiranom u poglavlju 3, pri čemu se koristi determinističan tok obrade slika, promjena veličine, središnji izrez na 300×300, normalizacija na ImageNet statistiku, bez dodatnih augmentacija. Time se osigurava da su mjerenja usporediva kroz epohe i različita pokretanja. Primarna metrika je točnost, računata na razini epohe kao omjer ispravno klasificiranih uzoraka i ukupnog broja uzoraka u validacijskom skupu. Paralelno se bilježi i gubitak na validaciji, što omogućuje uvid u dinamiku učenja i pojavu mogućeg pretreniranja.

Tijek treniranja strukturiran je u cikluse „train potom val“, a nakon svake validacijske faze ažurira se interni maksimum točnosti. Ako je postignuta nova najbolja vrijednost, model se smatra „najboljim do sada“ i njegove se težine čuvaju u memoriji, u suprotnom se uvećava brojač epoha bez poboljšanja. Rano zaustavljanje aktivira se kada broj uzastopnih epoha bez rasta točnosti dosegne zadani prag, čime se treniranje prekida i model vraća na najbolji zabilježeni skup težina, a isječak koda koji implementira rano zaustavljanje nalazi se na slici Slika 4‑1. Ovakav protokol sprječava nepotrebnu potrošnju računalnih resursa i smanjuje rizik regresije performansi pri kraju treniranja.



Slika 4‑1 Implementacija ranog zaustavljanja

Rezultati se sustavno dokumentiraju artefaktima u direktoriju TrainingFeedback. Grafički prikaz krivulja učenja, sadrži trenirajući i validacijski gubitak te validacijsku točnost po epohama, što vizualno prikazuje konvergenciju, potencijalno razilaženje gubitaka i trenutak aktivacije ranog zaustavljanja. Uz to se generira tekstualni zapis točnosti po epohama, ValAccuracyPerEpoch.txt, koji omogućuje preciznu identifikaciju epohe s najvećom vrijednošću i lakše usporedbe između različitih pokretanja. Težine konačnog modela serijaliziraju se u Models/ModelWeights/plant\_classifier\_efficientnetb3.pth, a taj se model u idućim odjeljcima koristi kao polazište za izvoz u mobilni format.

S obzirom na neravnomjernu zastupljenost klasa u skupu, uz ukupnu točnost je korisno po potrebi dopuniti izvještavanje metrikama osjetljivim na balans klasa, npr. makro-F1, i matricom zabune za kvalitativnu analizu tipičnih pogrešaka. Te se dopune, kada su izračunate, prikazuju u odjeljku o rezultatima.

## Rezultati treniranja

Tijek učenja zabilježen u ValAccuracyPerEpoch.txt pokazuje stabilan rast validacijske točnosti u ranim epohama, od 0,6814 (ep. 1) do 0,8302 (ep. 8), nakon čega slijedi faza finog podešavanja s manjim poboljšanjima. Najveća vrijednost postiže se u epohi 10 s 0,8465. U sljedećim epohama bilježe se oscilacije bez nadmašivanja maksimuma, 0,8370 u ep. 11; 0,8310 u ep. 12; 0,8344 u ep. 13; 0,8391 u ep. 14. S obzirom na postavljeni patience = 4, rano zaustavljanje aktivira se nakon epohe 14, četiri uzastopne epohe bez poboljšanja u odnosu na ep. 10, a model se vraća na najbolje težine iz epohe 10 koje se serijaliziraju za daljnju uporabu. Tijek promjene vrijednosti prametara može se vidjeti na slici Slika 4‑2.



Slika 4‑2 Krivulje učenja: trenirajući i validacijski gubitak te validacijska točnost po epohama.

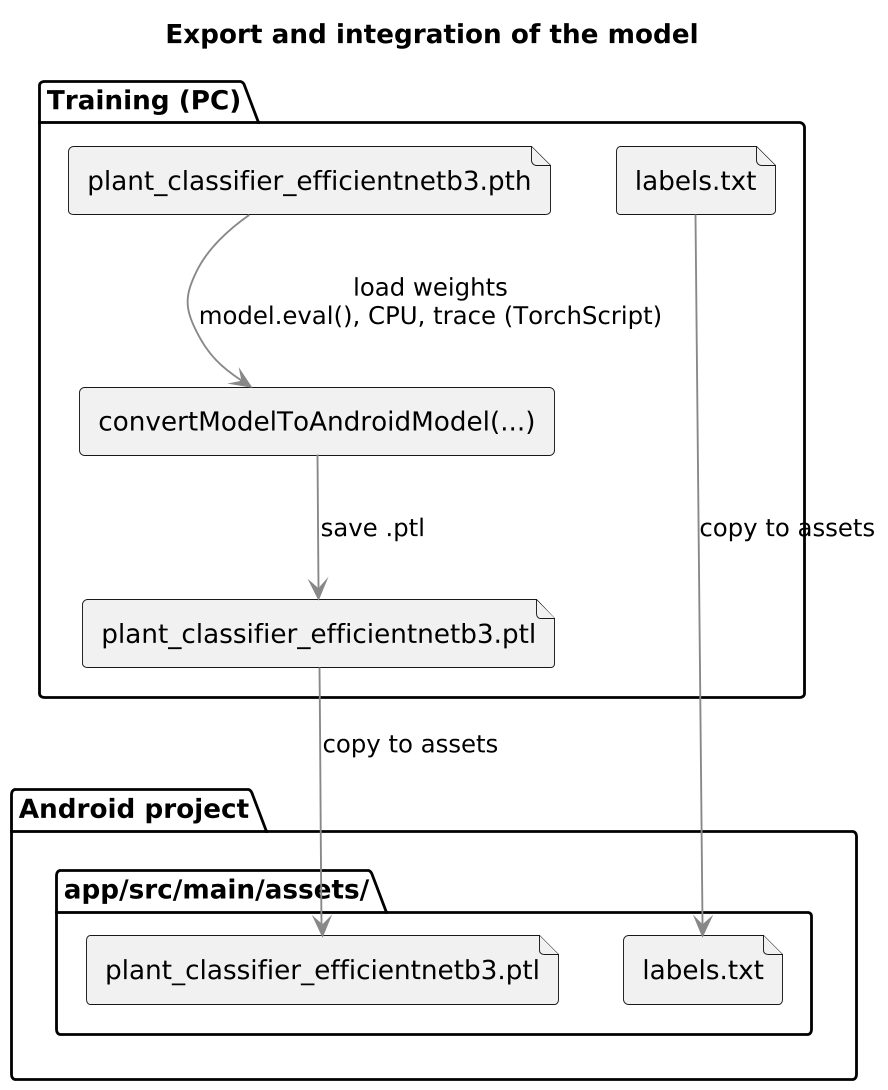
Krivulje učenja, trenirajući i validacijski gubitak te validacijska točnost po epohama, upućuju na očekivano ponašanje: kontinuirani pad trenirajućeg gubitka, početni pad i stabilizacija validacijskog gubitka te porast točnosti do plato-vrijednosti. To sugerira da je kombinacija arhitekture EfficientNet-B3 i primijenjenih augmentacija prikladna za zadatak klasifikacije 74 vrsta na skupu od 54253 slika. Ipak, pogreške su učestalije kod vizualno sličnih vrsta i u otežanim uvjetima, okluzije, mali objekt u kadru, složena pozadina, što je uobičajen obrazac u praktičnim klasifikacijskim problemima. Neravnomjerna zastupljenost klasa može blago favorizirati češće klase, taj se učinak može dodatno adresirati ponderiranjem gubitka ili ciljanim obogaćivanjem podataka u budućem radu.

U ovoj fazi, zaključujemo da treniranje daje stabilan „najbolji“ model s najvećom val. Točnošću od 0,8465 u ep. 10. Rano zaustavljanje sprječava pretreniranje i nepotrebne epohe, a odabrani checkpoint služi kao polazište za konverziju u mobilno-prikladan format i integraciju u Android aplikaciju u sljedećim odjeljcima.

## Izvoz modela i pripreme za Android

Nakon dovršetka treniranja i odabira „najboljih” težina, model se pretvara u mobilno-prikladan artefakt i zajedno s mapom oznaka integrira u Android aplikaciju. Izvoz je realiziran funkcijom *convertModelToAndroidModel*, koja učitava spremljene težine, prebacuje mrežu u evaluacijski način i na CPU, te generira *TorchScript* trace nad fiksnim ulazom. Dobiveni artefakt sprema se kao datoteka *Models/AndroidModel/plant\_classifier\_efficientnetb3.ptl*, a zatim se kopira u *Application/app/src/main/assets/plant\_classifier\_efficientnetb3.ptl* kako bi bio dostupan unutar aplikacije. Time se osigurava da je izvršni model dostupan iz assets direktorija bez potrebe za mrežom i bez dodatne instalacije na uređaju.

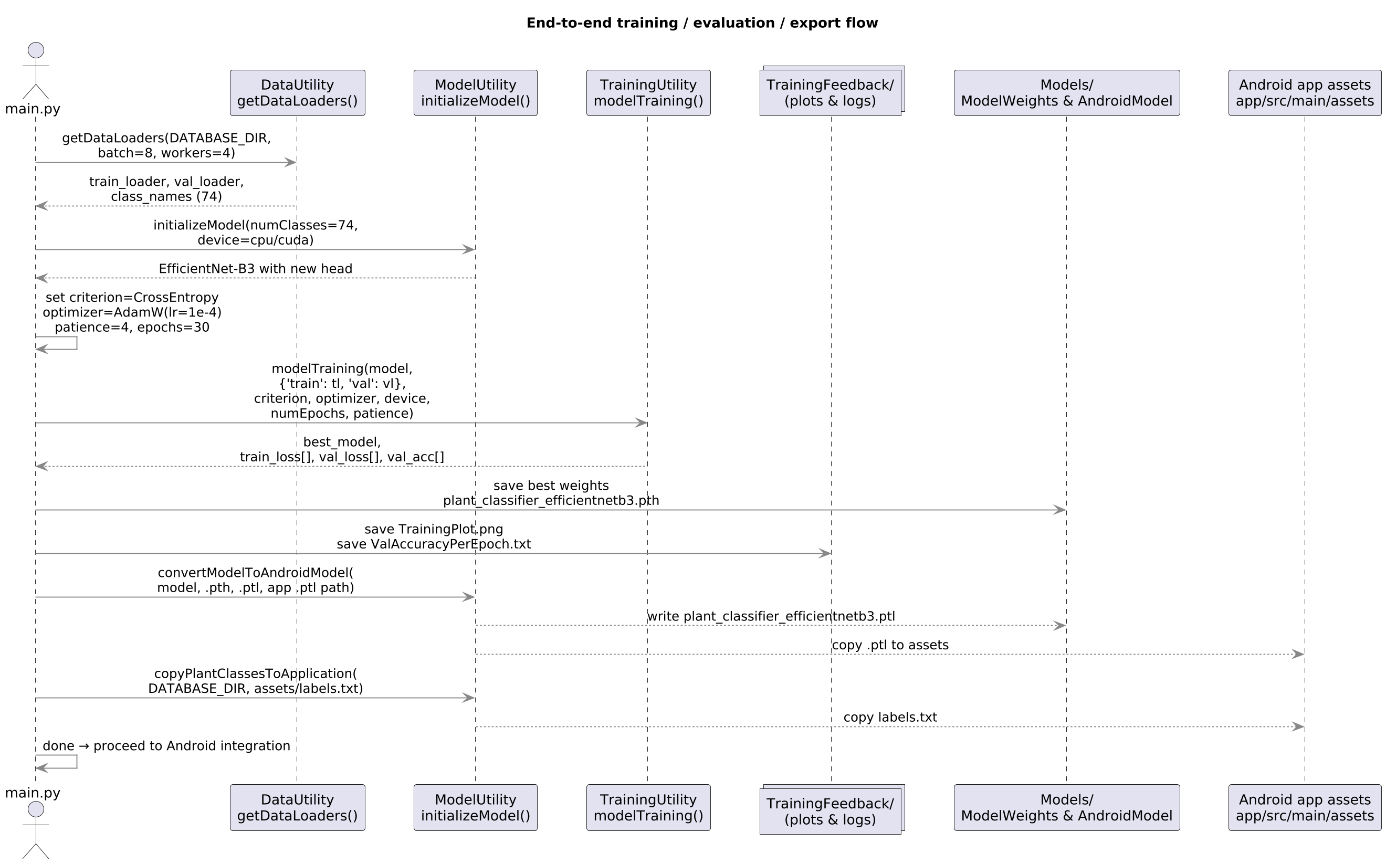
Uz model, aplikaciji je potrebna i dosljedna mapa klasa. Funkcija *copyPlantClassesToApplication* kopira popis oznaka, generiran tijekom izrade baze i treniranja, u *Application/app/src/main/assets/labels.txt*. Redoslijed ovih oznaka mora odgovarati redoslijedu na izlazu modela, čime se održava jednoznačno preslikavanje indeks, naziv vrste u svim fazama. Ovakva organizacija jamči da će prikazani rezultat u aplikaciji koristiti iste nazive i isti poredak kao i pri evaluaciji na računalu. Cijeli proces izvoza i prebacivanja u Android model mogu se vidjeti na slici Slika 4‑3.



Slika 4‑3 Izvoz modela i prebacivanjeu Android model

U kontekstu on-device inferencije, očekivanja modela nad ulazom prate trening pipeline: slika se pretvara u RGB, skalira/izrezuje na kvadratnu ciljnu rezoluciju, normalizira na *ImageNet* statistiku, *mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]*, a zatim se, kao tenzor oblika *1×3×H×W* tipa *float32*, prosljeđuje *TorchScript* modulu. Izlazi su sve 74 klase, a tijekom inferencije u aplikaciji tipično se primjenjuje softmax kako bi se dobile vjerojatnosti, izabire se top-1 ili top-k i taj se indeks preslikava u naziv klase čitanjem iz *labels.txt*. Ovakav pristup omogućuje potpuno offline izvođenje, čuva privatnost korisnika na način da slika ne napušta uređaj i eliminira latenciju povezanu s mrežnim pozivima.

Proces izrade mobilnog artefakta usklađen je s praktičnim zahtjevima integracije na Androidu, model je serijaliziran u *TorchScript Lite* formatu , .ptl formatu, prikladnom za *PyTorch Mobile runtime*, smješten je u *assets,* s ciljem pojednostavljenja distribucije i učitavanja, a pripadajuća datoteka s oznakama osigurava semantičku dosljednost rezultata. Dodatne optimizacije poput kvantizacije radi smanjenja memorijskog otiska i latencije predstavljaju prirodnu nadogradnju i moguće su u kasnijim iteracijama bez promjene sučelja aplikacije. U sljedećem poglavlju fokus se premješta na integraciju modela u Android aplikaciju, uključujući pretprocesiranje slika, pozivanje *TorchScript* modula i prezentaciju rezultata korisniku. Cjelokupni tijek skripte objasnjene kroz prethodne odjeljke nalazi se na slici Slika 4‑4.



Slika 4‑4 Tijek skripte za treniranje, evaluaciju i izviz modela

## Ograničenja i potencijalna poboljašanja

Trenutni tijek daje stabilan rezultat, ali ima nekoliko praktičnih ograničenja vezanih uz implementaciju. Skup augmentacija u DataUtility.py je namjerno „jak“, rotacije, affine/perspektiva, vertical flip, ColorJitter, RandomErasing, što povećava robusnost, ali može uvesti distribucijski pomak u odnosu na realne uvjete korištenja aplikacije. U idućim iteracijama smisleno je prilagoditi intenzitet transformacija onome što će korisnici najčešće snimati, npr. smanjiti vertikalna zrcaljenja ako nisu prirodna za većinu biljnih uzoraka.

U *TrainingUtility.py* treniranje se oslanja na fiksni korak učenja, Adam, i rano zaustavljanje s *patience=4*, uz praćenje točnosti kao glavne metrike. Ovakav protokol je jednostavan i učinkovit, ali može stati prije potencijalnog poboljšanja. U praksi se kao nadogradnja često koristi raspored stope učenja, npr. ReduceLROnPlateau, ili kratka druga faza finog podešavanja s manjom stopom učenja.

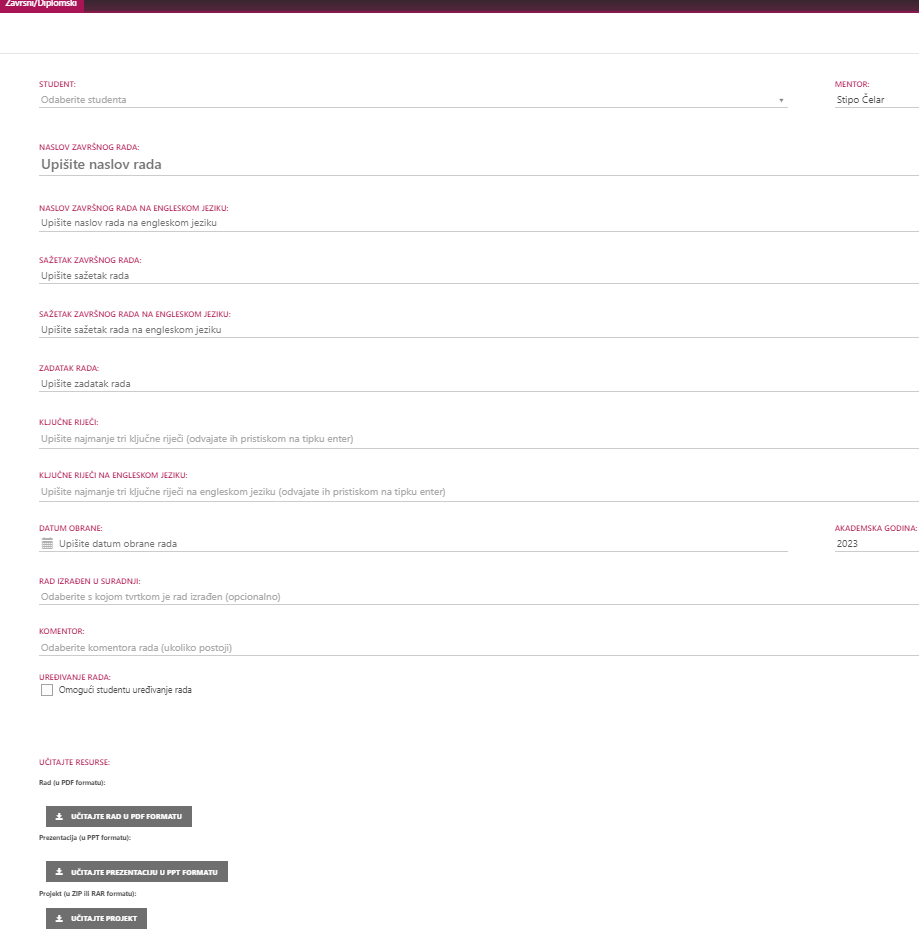
# Android aplikacija

Obvezni dijelovi radnje su slijedeći:

* Naslovnica
* Zadatak diplomskog rada
* Uvod
* Povezana istraživanja – Što je o problemu poznato?
* Teorijska razrada problema
* Primjer primjene (*Case study*) – ako je moguće
* Rezultati i njihov komentar
* Zaključak i budući rad
* Literatura
* Sažetak (*Abstract*)
* Ključne riječi (*Keywords*)
* Prilozi

**NASLOVNICU MOŽETE NAPRAVITI PREMA OVOM DOKUMENTU ALI JE PREPORUČLJIVO USPOREDITI JU NAKON TOGA S UPUTAMA S PORTALA MERLIN.**

**Obrazac 'Zadatak' priprema mentor i ON je odgovoran za njegovu ispravnost a za ostale dijelove radnje se možete poslužiti ovim dokumentom kao PREDLOŠKOM.**



Slika 4‑5 Sučelje FESB-ovog internog sustava za unos radnji

## Napomene za pisanje nekih dijelova radnje

## O temi/problemu

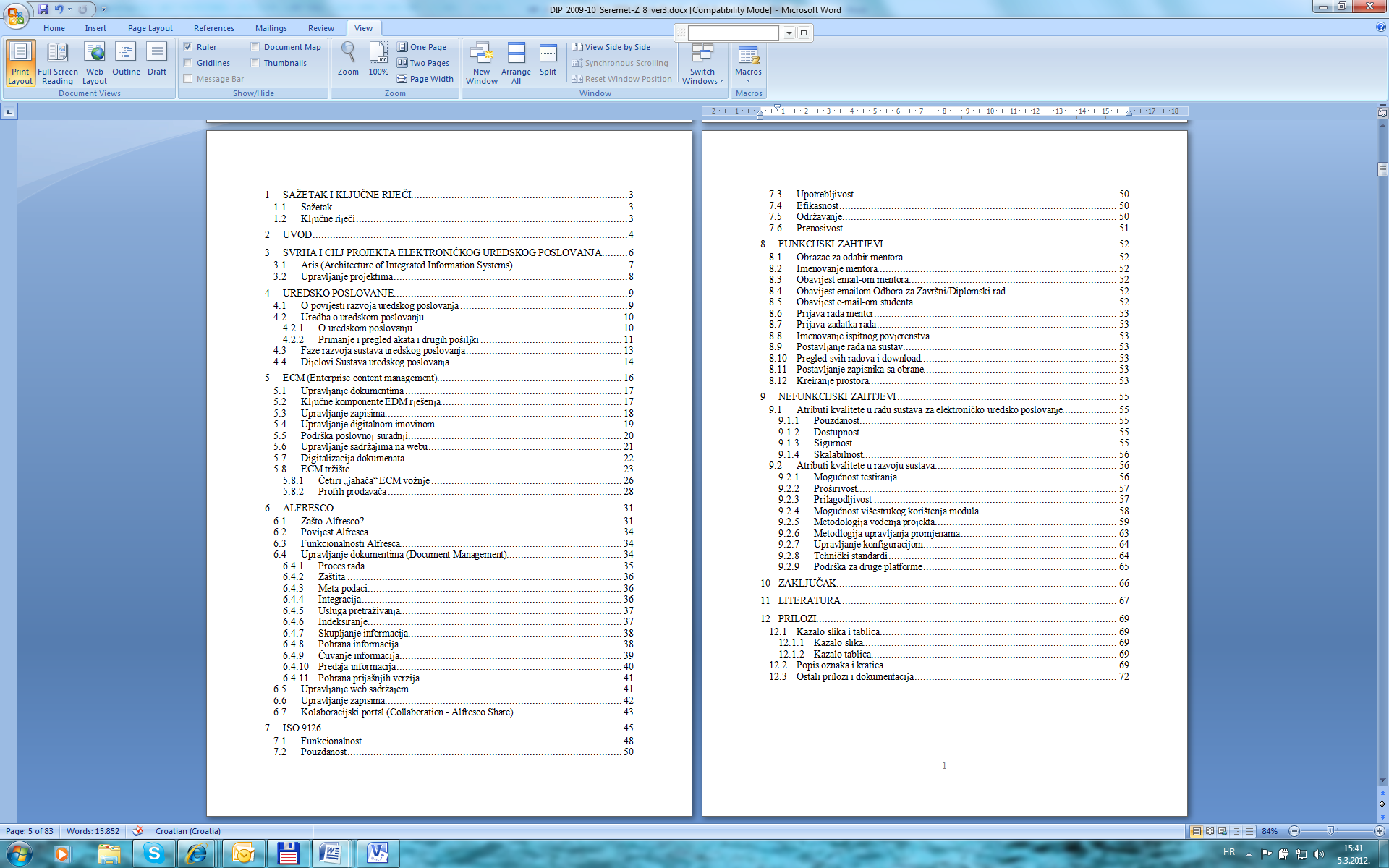
Što je problematika (tematika)?

* Zašto je zadani problem u stvari problem?
* Što je rješenje?
* **Zašto** je to rješenje zadanog problema?
  + jasna definicija problema koji se rješava/opisuje
  + definirati ga u:
    - naslovu
      * pročita ga 100% onih koji vide radnju
    - sažetku i ključnim riječima (5-6 ključnih riječi)
      * pročita ga 85% onih koji vide radnju
    - uvodnom poglavlju
      * pročita ga 14% onih koji vide radnju
  + razraditi ga u osnovnom dijelu radnje
    - * pročita ga 1% onih koji vide radnju
* Što su drugi rekli o tom problemu?
  + pregled literature
  + da se ne “otkriva topla voda”
  + S REFERENCAMA, a ne *copy paste*
* Što ja kažem o tome?
  + opis uvjeta, metoda rada, tehnički resursi...
* Koji je **rezultat** moga rada?
  + brojčani rezultati su dobrodošli
* Daljnji koraci/mogućnosti/radovi?
  + ...koji mogu slijediti iz radnje
  + donose se u poglavlju ZAKLJUČAK

## O strukturi radnje

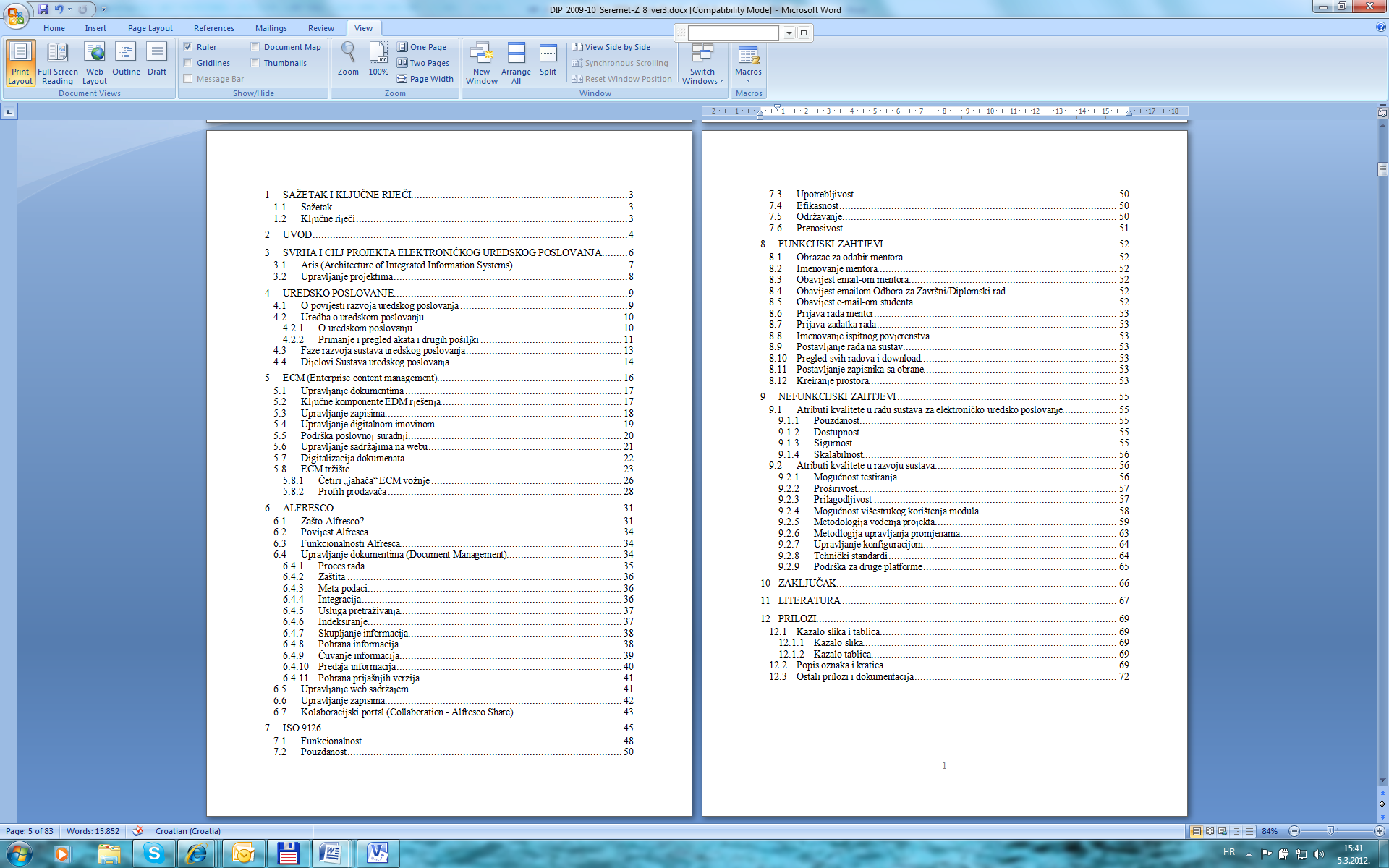
* Struktura radnje
  + Sažetak i ključne riječi
    - 10-ak rečenica o gore nabrojanom
    - od problema do rješenja i daljnjih radova (odnosno mogućnosti)
    - *5-6 ključnih riječi (mogu biti i složeni pojmovi) koji najbolje odražavaju sadržaj radnje*
  + Uvod
    - predstavljanje teme
    - opis centralnog problema
    - povezivanje teme/promlema s općenitim kontekstom
      * (tj. svrha, doprinos ili korisnost rješavanja tog problema – ako već nije spomenuto)
  + Razrada teme
    - Iz literature
      * Prikaz spoznaja o problemu do koji se došlo na temelju literature (*obvezan dio*)
    - Vlastiti doprinos
      * Praktični rad (izrada modela, SW,...) ili temeljitije istraživanje i usporedba nekih metoda, modela, SW, tvrtki,...
      * Odgovoriti na ključna pitanja: **Zašto/kako** predloženo rješenje rješava postavljeni/opisani problem
  + Zaključak
    - Postignuti rezultat
    - Koji su daljnji ciljevi (mogućnosti)?
  + Literatura
  + Prilozi
    - Popis oznaka i kratica
    - Popis slika
    - Popis tablica

Primjer sadržaja diplomskog rada prikazan je na donje dvije slike.



Slika 4‑6 Primjer sadržaja diplomskog rada (1/2)

Oznaku i naziv **slike** potrebno je staviti ispod slike (v. gornji primjer).



Slika 4‑7 Primjer sadržaja diplomskog rada (2/2)

# PLAN KOMUNIKACIJE

* Metoda rada i pisanja radnje
* Dogovor oko početnih obrazaca za pisanje radnje
* Razno

## Komunikacija na projektu

Mail adrese: mogu se koristiti i nefesbove domene, samo pošiljate mora biti jednoznačno i profesoru lako prepoznatljiv – inače neće odgovoriti.

## Mail-adrese i telefonski brojevi

Tablica 2 Popis mail-adresa i telefonskih brojeva sudionika projekta 2012-13 (FESB)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Osoba | mail | telefon |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

## Nazivi dokumenata

Radnje obvezno označavajte na slijedeći način:

* 250\_2009-10\_Prezime-Ime\_vx\_maticni-broj\_Naziv-radnje.doc(x)
* 120\_2009-10\_Prezime-Ime\_vx\_maticni-broj\_\_Naziv-radnje.doc(x)

npr: **250\_2009-10\_Volf-M\_xyz-20xz\_Cloud-sustavi.pdf**

pri čemu su oznake slijedećeg značenja:

* 250 diplomska radnja na diplomskom studiju 250 (Računarstvo)
* 120 završni rad na preddiplomskom studiju 120 (Računarstvo)
* Naziv-radnje skraćeni naziv radnje
* XXX brojčana oznaka studijskog usmjerenja na fakultetu
* Matični-broj matični broj studenta na fakultetu.

Izvješća i evidencije imaju u nazivu i oznaku razdoblja (od-do ili mjeseca) na koje se odnose:

* 250\_2009-10\_Prezime-Ime\_maticni-broj\_**izvj\_ozu-tra**\_Naziv-radnje.doc(x)
* 120\_2009-10\_Prezime-Ime\_**izvj\_tra**\_Naziv-radnje.xls(x)

(xls-evidencija uvijek treba biti kumulativna, tj. UKUPNA EVIDENCIJA OD POČETKA PROJEKTA.)

Primjer ispravno označene *diplomske radnje* studenta *Boška Antića* s temom *izrade aplikacije za uredsko poslovanje* bio bi:

**250\_2009-10\_Antic-Bosko\_ver2\_Uredsko-poslovanje.docx.**

U nazivu fileova je potrebno izbjegavati slova č, ć , ž, š i đ.

## Pitanja

Tablica 3 Bitna ili nebitna pitanja za komunikaciju

|  |  |  |
| --- | --- | --- |
|  | ***Pitanje – problemčić*** | ***Odgovor*** |
|  | Dobijanje povratne informacije | Kad god je moguće. U oba smjera. |
|  | Čuvanje povijesti | Čuvanje mailova se preporuča. I odgovaranje kao *replay.* |
|  | Konzultacije | Unaprijed se dogovaraju zajednički sastanci. Mailom student može najaviti potrebu i pojedinačnih konzultacija. |
|  | Kako često su sastanci? |  |

# PLANIRANJE I EVIDENCIJA

## IT-tvrtke u Hrvatskoj

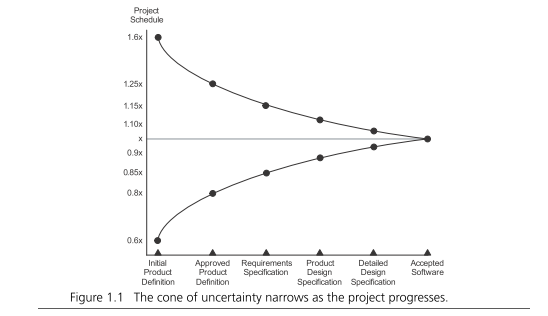
Prema podacima iz 2008. godine:

* 2,329 IT-tvrtki u CRO
* 90% ≤ 15 djelatnika

A profesor na diplomskom ima 5 diplomanata (svaki po 30 ECTS, tj. 50x30x30=4.500 h) i 5 na završnom (5x12x30=1.800 h). Dakle, prosječna IT-tvrtka u Hrvatskoj.

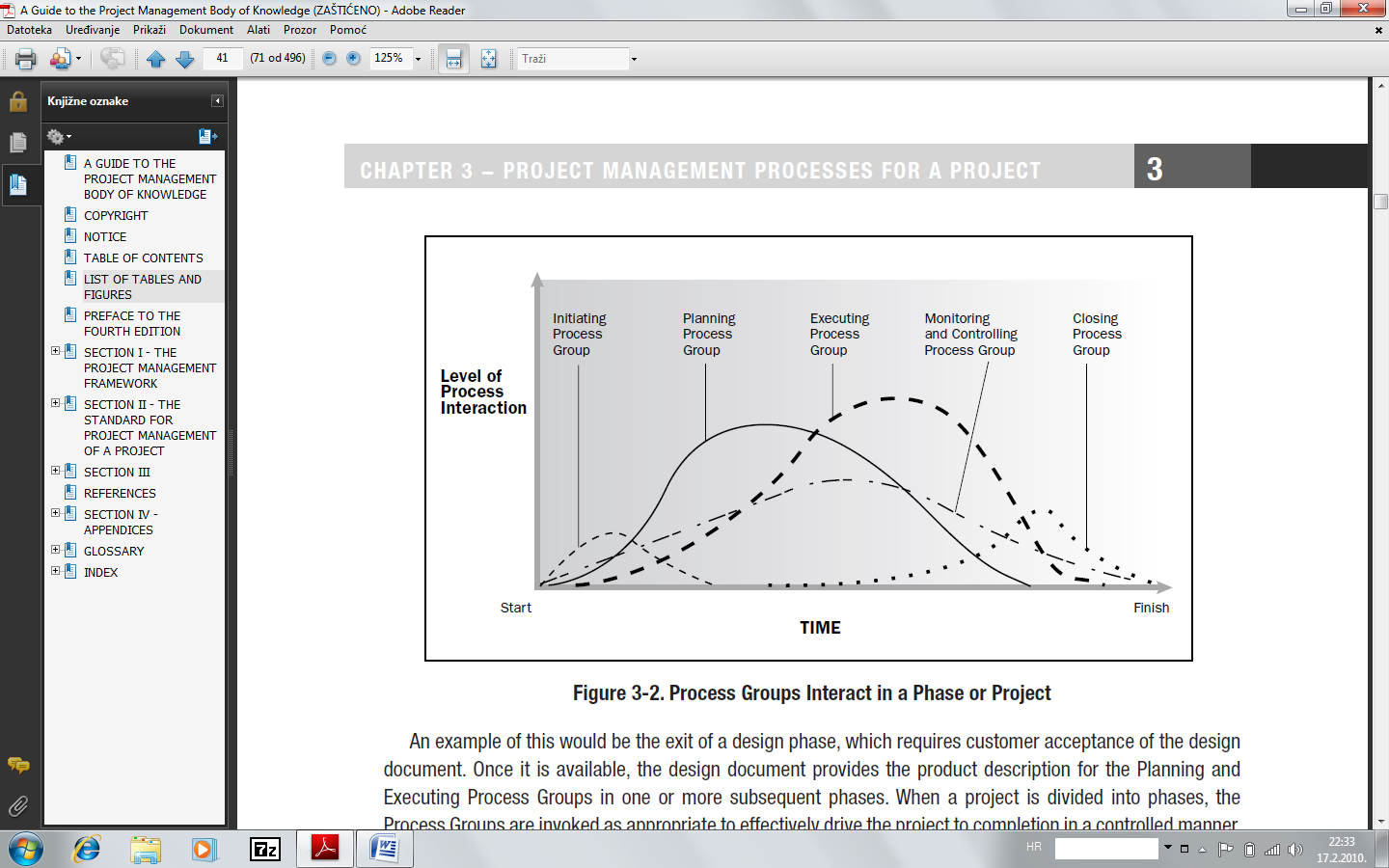
## Planiranje aktivnosti

Ovako izgleda preciznost planiranja na projektu.



Slika 6‑1 Konus nepouzdanosti [7]

Pogreška se dakle kreće od početnih cca (0,5-2) puta do +/-  10-20% nakon detaljne specifikacije dizajna (ali je to nakon što je proizvod/projekt 80% gotov!!!).



Slika 6‑2 Grupe projektnih aktivnosti tijekom projekta [7]

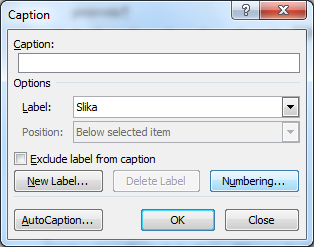
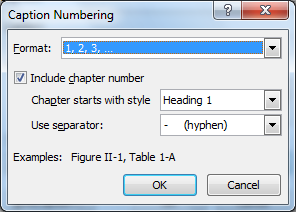
U xls tablici (***250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.xlsx***) nalazi se popis svih aktivnosti potrebnih na projektu (DIPLOMSKOM). Imate i zadanu količinu radnih dana (tj. sati). Računajte 30 ECTS x 30 h/ECTS = 900 h. Uzmite da je vaša planska vrijednost x=750 h (po osobi u timu – ako radite timski projekt) i napravite vremenski dijagram onih aktivnosti iz tablice.

Ovakva računica vam ostavlja još 20% rezerve u planu. A prva verzija plana ne mora u detalje precizirati baš sve aktivnosti iz **kasnijih** faza (dovoljno je to na višoj razini planirati u prvoj verziji tek da se vidi ukupna slika opterećenja i **ovisnosti** poslova).

Zato se nemojte ustručavati napraviti plan.

## Označavanje slika, tablica i ostalih objekata

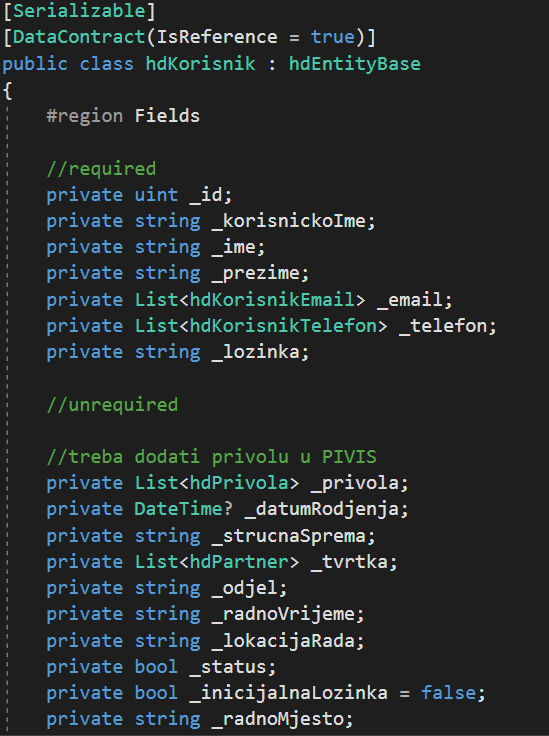
Slike se numeriraju automatski (opcija References/Insert caption). Odabere se label SLIKA i konfigurira prema donjim dvjema slikama:

Slika 6‑3 Konfiguriranje labele za slike

Na isti način se označavaju i ostali objekti (tablice, listinzi, napomene,...).

Ako u radnji imate kodove, onda izaberete *labelu* KOD s gornje slike – vidi donji primjer



Kod 4‑1 Primjer modela u WCF servisu

Popis svih kodova napravite u prilozima na kraju radnje (samo osvježite postojeći *Popis kodova*).

## Evidencija aktivnosti

Tablica 4 Evidencija rada studenta na projektu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | datum (dd.mm.gggg) | opis rada | kategorija | trajanje (30 i više min) | osoba |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| ... |  |  |  |  |  |

Popis aktivnosti na projektu:

* + - * 1. planiranje projekta DIPLOMSKI RAD (analize projekta, planiranje)
        2. čitanje literature
        3. analiza zahtjeva
        4. projektiranje (logički i fizički dizajn)
        5. programiranje
        6. testiranje
        7. sistem integracija (instalacija, konfiguriranje)
        8. testiranje sustava (SW+HW)
        9. pisanje dokumentacije za SW
        10. radnja pisanje
        11. konzultacije (stručni sastanci s mentorom)
        12. vođenje projekta (sastanci tima, zapisnici sastanaka,...)
        13. osiguranje kvalitete
        14. administracija (poslovi na referadi)
        15. ostalo (čekanja i sl.)

*Potrebno je pridržavati se ovih stavki u vođenju evidencije.*

Evidencija se vodi u Excel file (*250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.xlsx*) koji:

* se popunjava dnevno (par minuta) za aktivnosti koje su taj dan obavljene
* nije potrebno verzionirati jer je ovo kumulativna evidencija za cijeli projekt
* treba slati profesoru jednom tjedno kao izvješće o radu na projektu

# KONTROLA I NADZOR DIPLOMSKOG

O tijeku izrade radnje student(ica) će se konzultirati s profesorom u dogovorenim terminima (načelno 2-3 puta mjesečno).

**TJEDNO** student šalje profesoru mailom evidenciju o obavljenim tjednim aktivnostima (250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.xlsx).

**Mjesečno** student podnosi profesoru kratko pisano izvješće o:

* tijeku radnje
* planiranim i napravljenim aktivnostima u izvještajnom razdoblju
* planiranim aktivnostima u slijedećem izvještajnom razdoblju
* smetnjama u izvještajnom razdoblju i mjerama otklanjanja
* rizicima u slijedećem izvještajnom razdoblju

Uz mjesečno izvješće student dostavlja profesoru i:

* kumulativnu evidenciju rada (250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.xls; od početka projekta)
* ažuriranu verziju plana projekta
  + **250\_2013-14\_Prezime-Ime\_vX\_maticni-broj\_Naziv-radnje.mpp**
  + inicijalna verzija se izrađuje na početku projekta
  + plan se ažurira prema potrebi u dogovoru s profesorom
* aktualnu verziju RADNJE
  + 250\_2013-14\_Prezime-Ime\_vX\_maticni-broj\_Naziv-radnje.doc(x)
  + Kada profesor nešto korigira u radnji, onda on napravi ***podverziju*** poslane mu verzije (vX.1) i vrati je studentu mailom

# METODA RADA I PISANJA RADNJE

## Pisanje radnje

Radnja se počinje pisati ODMAH! U početku imate bar naslovnicu i naslove. Gradite sadržaj pomalo.

Pišite i brišite – nemojte se bojati da ćete nešto pogrešno napisati.

Planirajte najmanje tri verzije radnje koje mentor mora pogledati i dati sugestije (makar i na 5 minuta).

U mpp-planu isplanirajte te tri verzije i njihov pregled, finalni potpis i uvezivanje.

## ECTS

4 ECTS

(30 sati predavanja – 1 ECTS

15 sati vježbi – 0,5 ECTS

75 sati samostalnog učenja – 2,5 ECTS)

5 ECTS

(30 sati predavanja - 1 ECTS

30 sati vježbi - 1 ECTS

90 sati samostalnog učenja – 3 ECTS)

6 ECTS

(30 sati predavanja – 1 ECTS,

30 sati vježbi – 1 ECTS,

120 sati samostalnog učenja – 4 ECTS)

završni rad 12 ECTS 360 h

diplomski rad 30 ECTS 900 h

***OVA RAČUNICA JE RAĐENA ZA PROSJEČNOG STUDENTA.***

## Alati za pisanje

* Zotero
  + Alat za referenciranje literature i AUTOMATSKO kreiranje referencirane literature
  + v. How To Use Zotero (A Complete Beginner's Guide) – <https://www.youtube.com/watch?v=JG7Uq_JFDzE>
* MS Word
* MS Project
* MS Excel

## Oblik finalnog dokumenta (obvezno!)

* doc (ili docx)
* pdf
* mpp
* ppt (ili pptx)
* xls(x)

Radnja i prezentacija (obje u pdf obliku) objavljuju se na portalu Fakulteta a od prije par godina i na javno dostupnom Nacionalnom repozitoriju završnih i diplomskih radova DABAR (<https://zir.nsk.hr/>).

# DOGOVOR OKO POČETNIH OBRAZACA ZA PISANJE RADNJE

## Neka to bude ovaj dokument. (heading 2)

Treba samo umjesto ovog teksta pisati svoj tekst u stilu NORMAL i ostalim stilovima i kazalima iz ovog dokumenta.

### On sadrži potrebne stilove. (heading 3)

Normal

Tablica 5 style caption

Slika 9‑1 style caption (isti za slike i tablice)

1. Početi pisati literaturu od samog početka pisanja radnje
2. Style Literatura Sdajafljdfjdf dfjdg Početi pisati literaturu od samog početka pisanja radnje jkf kPočeti pisati literaturu od samog početka pisanja radnje Početi pisati literaturu od samog početka pisanja radnjeg Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g gkjkg ggg
3. Style Literatura Sdajafljdfjdf dfjdg Početi pisati literaturu od samog početka pisanja radnje jkf kPočeti pisati literaturu od samog početka pisanja radnje Početi pisati literaturu od samog početka pisanja radnjeg Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g gkjkg ggg
4. Style Literatura Sdajafljdfjdf dfjdg Početi pisati literaturu od samog početka pisanja radnje jkf kPočeti pisati literaturu od samog početka pisanja radnje Početi pisati literaturu od samog početka pisanja radnjeg Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g gkjkg ggg
5. Style Literatura Sdajafljdfjdf dfjdg Početi pisati literaturu od samog početka pisanja radnje jkf kPočeti pisati literaturu od samog početka pisanja radnje Početi pisati literaturu od samog početka pisanja radnjeg Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g Početi pisati literaturu od samog početka pisanja radnjedf g gkjkg ggg

### Upute za napredno korištenje Worda

Za one koji žele naučiti nešto više o formatiranju dokumenata u Wordu korisne će biti detaljne slikovne upute na stranici <http://www.shaunakelly.com/word>.

# NAPOMENE

Ako je zadatak radnje IZRADA aplikacije, onda je NUŽNO u radnji priložiti dokaz da je aplikacija NAPRAVLJENA – osim izlista koda stavite i scren shot te na prezentaciji pokažite pred povjerenstvom kako aplikacija radi.

# NAPOMENE PROFESORA

Kada nešto komentiram u radnji, onda najčešće koristim oblik komentara ispod paragrafa poput ovoga dolje.

prof 4 Komentar koji se odnosi na gornji paragraf, sliku, tablicu i sl.

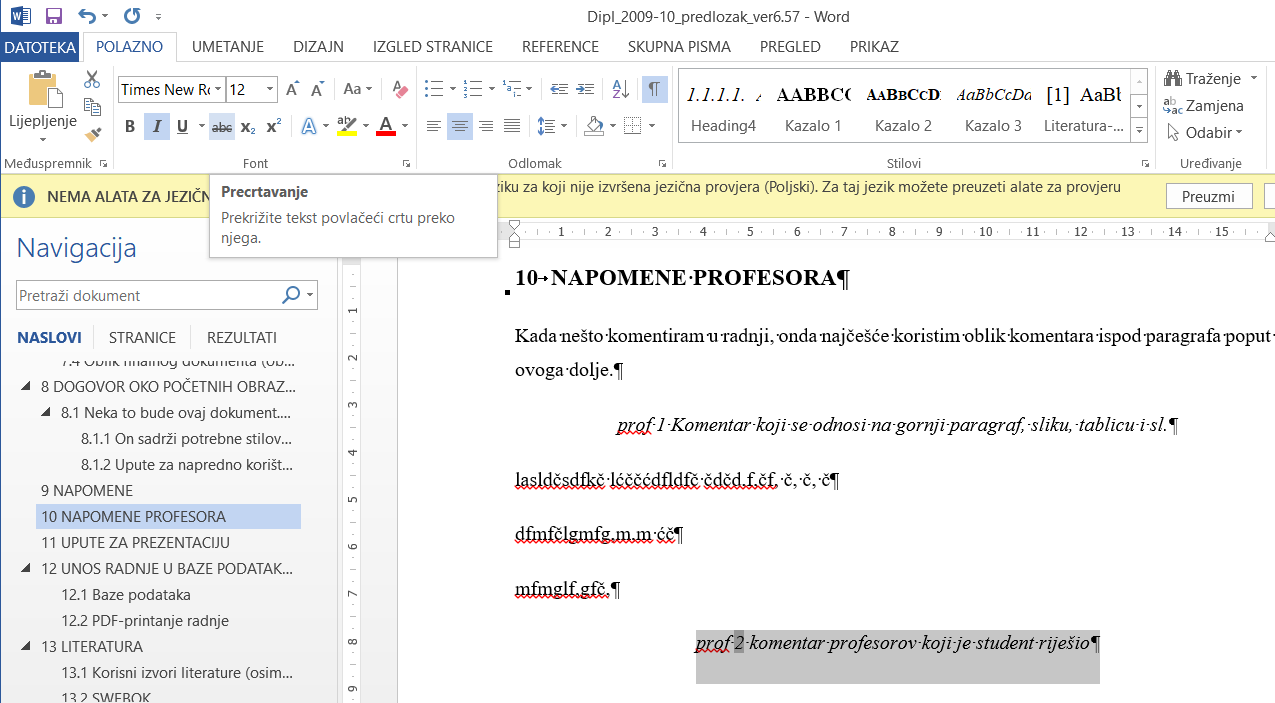
lasldčsdfkč lćččćdfldfč čdčd,f,čf, č, č, č

dfmfčlgmfg,m,m ćč

mfmglf,gfč,

~~prof 5 Komentar profesorov koji je student riješio~~

Ako je komentar riješen, onda ga treba PRECRTATI kao na donjoj slici (Slika 9‑1).



Slika 11‑1 Precrtavanje riješenog komentara

Ako profesor stavi napomene (*notes* – v. tab *Review*), nemojte ih brisati nego postupite prema njima a eventualne odgovore napišite u njima.

Te *notes* će obrisati profesor kada budu potpuno riješene.

# UPUTE ZA PREZENTACIJU

Dogovoreno trajanje prezentacije je 15 minuta i 10-15 minuta za pitanja mentora i ostalih članova povjerenstva (i publika može pitati – obično ona nema pitanja).

Kada sam ja član povjerenstva i slušam prezentacije, nađem se često u nedoumici: ne mogu na osnovu kandidatove prezentacije zaključiti slijedeće:

1. Što je PROBLEM koji se rješava (rješavao) u okviru diplomskog
2. Povijest tog problema (tj. kratki uvod u taj problem) veza s drugim tehničkim pojmovima/područjima/problemima koje razumije netko tko nije na tom problemu radio pola godine (tj. koliko i kandidat kad je o njemu počeo razmišljati na početku razgovora s mentorom)?
3. Koja je ključna literatura od koje je krenuo i koju je referencirao u radnji (kopiran popis na jedan slide i kratki komentar ključnih stavki kroz rečenicu-dvije) – tj. što su drugi rekli o tom problemu?
4. Koja je metoda korištena u radu – ovo je možda najuočljivije (vidi se je li programirao, ili samo teoretski istraživao literaturu, radio model u laboratoriju)
5. Koje su tehnologije, STANDARDI, MODELI korištene – rijetko se naglasi (osim negdje usput)?
6. Šta je sam kandidat napravio (ovo neki od članova pitaju u svojim pitanjima jer im ne bude jasno u uvodnom izlaganju)?
7. Gdje bi se to moglo primijeniti? Ima li mogućnosti neke primjene? Što dalje kad je ovo riješeno? Mogu li se time rješavati neki drugi slični problemi?

Kada nisam mentor, nastojim prije obrane pronaći ove odgovore u UVODU, ZAKLJUČKU i KAZALU.

Kada sam mentor, onda znam odgovore na ova pitanja pa nastojim s 2-3 pitanja otvoriti prostor za diskusiju kolegama iz povjerenstva (jer mentor prvi pita).

Stoga bi bilo dobro da vi u svojih 15-ak minuta nastojite sa 7-8 slideova dati odgovore na gornja pitanja.

AKO JE NETKO RADIO APLIKACIJU (praktični dio), ONDA TO I DEMONSTRIRA TIJEKOM OBRANE.

***I prezentacija se također prilaže na DVD-u uz finalnu radnju, nazvana kao i radnja (u pdf i ppt obliku bez oznake verzije):***

***250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.pdf***

***250\_2013-14\_Prezime-Ime\_maticni-broj\_Naziv-radnje.ppt(x)***

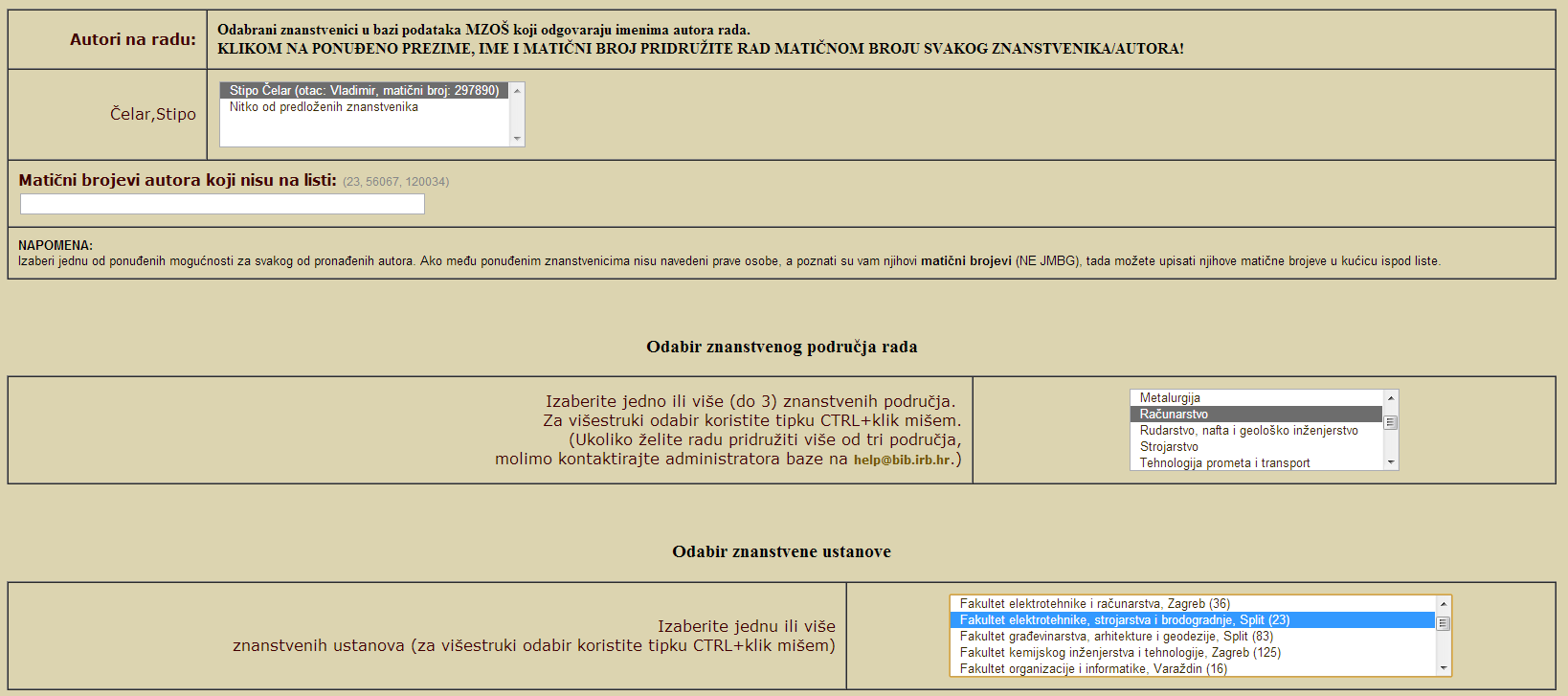
# UNOS RADNJE U BAZE PODATAKA I PDF-PRINT

## Baze podataka

Na stranici Instituta Ruđer Bošković nalazi se baza podataka u koju biste trebali unijeti podatke o svojoj radnji. Time će vaš rad 'ući u povijest'.

Na stranici <http://bib.irb.hr/lista-radova?autor=297890> imate opciju ***unos novih radova***.

Pri potvrđivanju unesenih podataka u bazu potrebno je izabrati/potvrditi slijedeći odabir sa donje slike:



Slika 13‑1 Potvrđivanje podataka o unesenom radu u BIB IRB

Pregled datuma diplomiranja po mentorima dostupan je na adresi <http://intranet.fesb.hr/Nastava/Zavr%C5%A1niidiplomskiradovi/tabid/995/Default.aspx>.

Krajnji rezultat unosa izgleda kao što je Mili napravio:

<http://bib.irb.hr/prikazi-rad?&rad=478826>

Ako vam treba pomoć kod pristupa stranici (registracija, AAI-account,…), obratite se na mail Žarku Rnjaku ([zarko.rnjak@fesb.hr](mailto:zarko.rnjak@fesb.hr)) – on će vam pomoći za tehničke probleme.

## PDF-printanje radnje

U word dokument trebate uključiti obrazac ZADATAK (koji dobijete od mentora neposredno prije uvezivanja radnje) i radnju printati u PDF (***U IMENU OSTAJU SVE GENERALIJE IZ POGLAVLJA O NAZIVU DOKUMENATA).*** *Error! Reference source not found.*).

Ovaj pdf uključite u CD koji predajete mentoru na kraju projekta.

RAZLOG: jednom će i ovi pdf-ovi završiti u bazama podataka i na portalu Fakulteta.

# UPUTE ZA LITERATURU

## Korisni izvori literature (osim Wikipedije)

<http://www.scopus.com>

<http://www.sciencedirect.com>

[*http://www.online-baze.hr/baze*](http://www.online-baze.hr/baze)

[*http://www.fer.hr/knjiznica*](http://www.fer.hr/knjiznica)

<http://www.online-baze.hr/ob/thomsonfree>

[*http://www.freeebooksearch.com*](http://www.freeebooksearch.com/)

<http://orion.fesb.hr/dotnetnuke/Znanostitehnologija/Onlinecasopisi>

[http://scholar.google.com](http://scholar.google.com/)

[www.sei.cmu.edu](http://www.sei.cmu.edu/)

## SWEBOK

Za potrebe SW-inženjerstva bitna literatura dostupna je na:

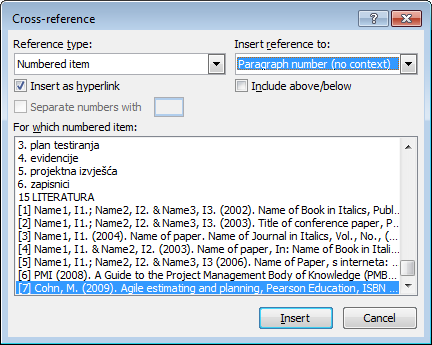
[www.swebok.org](http://www.swebok.org)

### Način pretraživanja

* odabir ključnih riječi
* odabir izvora
* odabir kriterija
* pretraživanje (i ČITANJE)
  + Paretto-princip (80-20)

### Navođenje izvora literature

* literatura se piše od početka pisanja radnje a ne na kraju („zadnju noć“) – takvu radnju mentor ne potpisuje („ima još rokova“ ☺)
* referencira se kao što je u ovim uputama [u uglatim zagradama]
  + koristiti opciju Worda: References – Cross-reference (vidi donju sliku)



Slika 14‑1 Uputa za citiranje literature

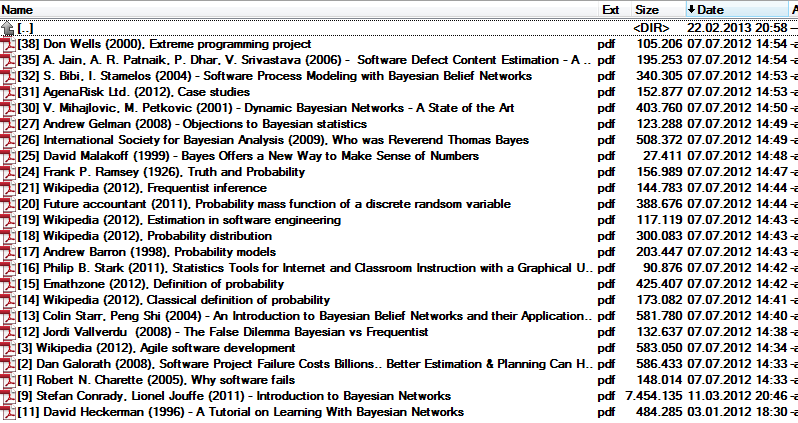
***Važna napomena:***

* POPIS LITERATURE I NJENO NAVOĐENJE U RADNJI TREBA ZAPOČETI ODMAH S POČETKOM, OD POČETKA A NE „NA KRAJU ĆU JA TO SREDITI“ ☺
* Koristite besplatni SW ZOTERO (<https://www.youtube.com/watch?v=JG7Uq_JFDzE>)
  + Ovaj alat automatizira citiranje literature i pravljenje POPISA citirane literature

### Literatura za mentora

***ZA MENTORA JE PREPORUČLJIVO ISPORUČITI SVU LITERATURU KOJU NAVODITE U RADNJI za koju mislite da može koristiti nekome od kolega koji će nastaviti rad na vašoj temi!***

***Literaturu numerirajte brojevima koje ste joj dali u radnji***

******

Slika 14‑2 Popis literature priložene uz finalnu radnju za mentora – primjer

Literatura se u radnji navodi pred kraj radnje u zasebnom poglavlju

# ZAKLJUČAK

U ovom poglavlju (na 1-2 stranice) potrebno je sažeti ključne rezultate do kojih se došlo tijekom istraživanja i/ili rada te što bi se moglo/trebalo dalje raditi / razvijati / istražiti / popraviti / unaprijediti..

LITERATURA

1. Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach. Pearson, 2020.
2. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.
3. IBM. “Supervised vs. Unsupervised Learning.” Dostupno na: <https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning>, zadnji pristup: 25.08.2025.
4. Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction. MIT Press, 2018.
5. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 1958.
6. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.
7. Krizhevsky, A., Sutskever, I., & Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. NeurIPS, 2012.
8. Szeliski, R. Computer Vision: Algorithms and Applications. Springer, 2022.
9. Gonzalez, R. C., & Woods, R. E. Digital Image Processing. Pearson, 2018.
10. Kamilaris, A., & Prenafeta-Boldú, F. X. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 2018.
11. Grinblat, G. L., Uzal, L. C., Larese, M. G., & Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. Computers and Electronics in Agriculture, 127, 418–424.
12. Hughes, D. P., & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv:1511.08060.
13. Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. Pattern Recognition, 71, 1–13.
14. Goëau, H., Bonnet, P., & Joly, A. (2017). Plant identification in an open-world (LifeCLEF 2017). CLEF Working Notes.
15. Affouard, A., Goëau, H., Bonnet, P., & Joly, A. (2017). PlantNet App in the Era of Deep Learning. ICCV Workshops.
16. Google. (2023). Google Lens. Dostupno na: <https://lens.google/>

Svaka stavka literature, navedena u gornjem popisu, mora u radu biti referencirana najmanje jednom. Stavke se ulistavaju u popis onim redoslijedom kojim se pojavljuju u radnji.

PRILOZI

Kazalo slika, tablica i kodova

Kazalo slika

[Slika 2‑1 Prikaz kako radi nadzirano strojno učenje. (preuzeto s https://maddevs.io/blog/semi-supervised-learning-explained/ ) 3](#_Toc207125542)

[Slika 2‑2 Prikaz kako radi nenadzirano strojno učenje (preuzeto s https://maddevs.io/blog/semi-supervised-learning-explained/) 4](#_Toc207125543)

[Slika 2‑3 Prikaz podržanog strojnog učenja (preuzeto s https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html) 5](#_Toc207125544)

[Slika 2‑4 Primjer izgleda višeslojnog perceptrona, tzv. MLP-a (preuzeto s https://learnopencv.com/understanding-convolutional-neural-networks-cnn/) 7](#_Toc207125545)

[Slika 2‑5 Izgled konvolucijskih slojeva u VGG-16 konvolucijskoj neuralnoj mreži (preuzeto s https://learnopencv.com/understanding-convolutional-neural-networks-cnn/) 8](#_Toc207125546)

[Slika 2‑6 Neuronska mreža trenirana od počeka (preuzeto s https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective) 9](#_Toc207125547)

[Slika 2‑7 Neuronska mreža nakon tranfer learning-a (preuzeto s https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective) 9](#_Toc207125548)

[Slika 2‑8 Sučelje FESB-ovog internog sustava za unos radnji 12](#_Toc207125549)

[Slika 2‑9 Primjer sadržaja diplomskog rada (1/2) 16](#_Toc207125550)

[Slika 2‑10 Primjer sadržaja diplomskog rada (2/2) 17](#_Toc207125551)

[Slika 4‑1 Konus nepouzdanosti [7] 20](#_Toc207125552)

[Slika 4‑2 Grupe projektnih aktivnosti tijekom projekta [7] 21](#_Toc207125553)

[Slika 4‑3 Konfiguriranje labele za slike 22](#_Toc207125554)

[Slika 7‑1 style caption (isti za slike i tablice) 28](#_Toc207125555)

[Slika 9‑1 Precrtavanje riješenog komentara 31](#_Toc207125556)

[Slika 11‑1 Potvrđivanje podataka o unesenom radu u BIB IRB 34](#_Toc207125557)

[Slika 12‑1 Uputa za citiranje literature 37](#_Toc207125558)

[Slika 12‑2 Popis literature priložene uz finalnu radnju za mentora – primjer 38](#_Toc207125559)

Kazalo tablica

[Tablica 3‑1 Popis mail-adresa i telefonskih brojeva sudionika projekta 2012-13 (FESB) 10](#_Toc161996528)

[Tablica 3‑2 Bitna ili nebitna pitanja za komunikaciju 11](#_Toc161996529)

[Tablica 4‑1 Evidencija rada studenta na projektu 15](#_Toc161996530)

[Tablica 7‑1 style caption 20](#_Toc161996531)

Kazalo kodova

[Kod 4‑1 Primjer modela u WCF servisu 14](#_Toc161996532)

Popis oznaka i kratica

CMM Capability Maturity Model

SW-CMM Software Capability Maturity Model

SE-CMM System Engineering Capability Maturity Model

Ostali prilozi i dokumentacija

Kazalo profesorovih napomena

[prof 1 Ova stranica se PRINTA BEZ potpisa a onda se POTPIŠE PLAVOM bojom (tako da potpis bude samo u pdf-verziji) 1](#_Toc161996533)

[prof 2 I ova stranica se PRINTA BEZ potpisa a onda se POTPIŠE PLAVOM bojom (tako da potpis bude samo u pdf-verziji) 2](#_Toc161996534)

[prof 3 Žutom bojom označeni tekst treba uskladiti (ILI je 'završni' ILI je 'diplomski' i skinuti boju) 2](#_Toc161996535)

[prof 4 Komentar koji se odnosi na gornji paragraf, sliku, tablicu i sl. 23](#_Toc161996536)

[~~prof 5 Komentar profesorov koji je student riješio~~ 23](#_Toc161996537)

Upravljanje izmjenama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OZNAKA  VERZIJE | DATUM | PRIPREMIO | ODOBRIO | KRATKI OPIS IZMJENA |
| 7.3 | 2024-03-22 | Stipe Čelar | Stipe Čelar | Razine studijâ na obrascu ZADATAK usklađene s novim ZVOZD-om (‘sveučilišni prijediplomski’ i sveučilišni diplomski’).  Iz galerije stilova obrisani nepotrebni stilovi  Pojašnjen način pisanja poglavja UVOD.  Isključeno potpisivanje elektroničke verzije rada.  Isključen dio koji je tražio dostavljanje materijala na CD-u. |
| 7.2 | 2023-03-01 | Stipe Čelar | Stipe Čelar | Stari logo Sveučilišta je zamijenjen novim |
| 7.1 | 2022-08-20 | Stipe Čelar | Stipe Čelar | U IZJAVI su naglašene riječi u ženskom i muškom rodu (žutom bojom) – potrebno je odabrati odgovarajući; U naslovnici i ZADATKU su naglašene riječi ZAVRŠNI i DIPLOMSKI (žutom bojom) – potrebno je eventualno odabrati odgovarajuću Istaknute su napomene za pisanje *sažetka* i *ključnih riječi* |
| 7.0 | 2022-03-18 | Stipe Čelar | Stipe Čelar | Numeriranje počinje tek od poglavlja UVOD; U IZJAVI su naglašene riječi ZAVRŠNI i DIPLOMSKI (žutom bojom) – potrebno je odabrati odgovarajuću |
| 6.9 | 2021-06-23 | Stipe Čelar | Stipe Čelar | Dodano poglavlje ZAKLJUČAK (razina 1); poglavlja ‘Upravljanje izmjenama’ i ‘Kazalo profeosorvih napomena’ stavljeni kao potpoglavlja u ‘Ostale priloge i dokumentaciju’ |
| 6.8 | 2020-07-27 | Stipe Čelar | Stipe Čelar | Uključene napomene za printanje sa i bez potpisa na ZADATKU i IZJAVI |
| 6.7 | 2020-05-11 | Stipe Čelar | Stipe Čelar | Korigirano poravnanje na naslovnici i prof. zvanja |
| 6.6 | 2020-01-27 | Stipe Čelar | Stipe Čelar | Napravljena zasebna hijerarhija oznaka za literaturu, priloge i sažetak – bez brojeva |
| 6.57 | 2019-01-16 | Stipe Čelar | Stipe Čelar | Napomene za ručno upisane informacije na CD-u  Dodao poglavlje o profesorovim komentarima |
| 6.56 | 2018-02-20 | Stipe Čelar | Stipe Čelar | Stavljena IZJAVA student/ice o samostalnom radu; odvajanje ključnih riječi **zarezom** (,) umjesto **točka-zarezom** (;) uključen je naslov radnje na engleskom na stranici s **abstractom** |
| 6.55 | 2015-03-05 | Stipe Čelar | Stipe Čelar | Stavljena adresa popisa diplomiranih studenata po mentorima i studijima |
| 6.54 | 2014-10-07 | Stipe Čelar | Stipe Čelar | Na početak ubačena naslovnica, omot CD-a i ZADATAK |
| 6.53 | 2014-09-19 | Stipe Čelar | Stipe Čelar | SAŽETAK je stavljen iza PRILOGA |
| 6.52 | 2013-11-29 | Stipe Čelar | Stipe Čelar | Manje dopune - |
| 6.51 | 2013-07-17 | Stipe Čelar | Stipe Čelar | Nazivi DIP i ZAV mijenjani u brojčane oznake studijskih osmjerenja (120, 250,...) |
| 6.5 | 2013-04-17 | Stipe Čelar | Stipe Čelar | Napravljen indent kod naslova  Sažetak i ključne riječi prebačeni s početka na kraj radnje |
| 6.4 | 2012-10-10 | Stipe Čelar | Stipe Čelar | *Pojašnjenje ranije manje jasnih detalja.* |
| 6.3 | 2012-03-05 | Stipe Čelar | Stipe Čelar | *Pojašnjenje ranije manje jasnih detalja.* |
| 6.2 | 2011-10-04 | Stipe Čelar | Stipe Čelar | *Upute za literaturu – automatsko numeriranje* |
| 6.1 | 2011-03-10 | Stipe Čelar | Stipe Čelar | *Manje izmjene.* |
| 6 | 2010-10-04 | Stipe Čelar | Stipe Čelar | *Unos radnje u bazu podataka (IRB) i pdf-print radnje sa zadatkom* |
| 5 | 2010-09-13 | Stipe Čelar | Stipe Čelar | *Literatura: s interneta i zadnji pristup* |
| 4 | 2010-08-25 | Stipe Čelar | Stipe Čelar | *Primjer završetka uvodnog poglavlja i upute za prezentaciju* |
| 3 | 2010-07-28 | Stipe Čelar | Stipe Čelar | *Sažetak i ključne riječi ostaju kao poglavlje ali bez numeraciju na razini potpoglavlja*  *Dodano poglavlje o označavanju slika i tablica.* |
| 2 | 2010-07-15 | Stipe Čelar | Stipe Čelar | *Dodano poglavlje o upravljanju promjenama. Uključena napomena s obrane o izostanku screen shotova app* |
| 1 | 2010-03-08 | Stipe Čelar | Stipe Čelar | Prva verzija poslana svim diplomantima |
| 0 | 2009 | Stipe Čelar |  | Inicijalna verzija. Nastala na laboratorijskim vježbama za potrebe seminara. |

U svojoj radnji napravite ovakvu tablicu sve do zadnje verzije – do printanja. Tada ovo stavite kao jedan od priloga (ako mislite da vam je to bitno) ili ga izbrišite.

Potrebno je priložiti i primjerke ostale dokumentacije (ako postoje):

1. plan upravljanja kvalitetom
2. projektni planovi
3. plan testiranja
4. evidencije
5. projektna izvješća
6. zapisnici

SAŽETAK/ABSTRACT I KLJUČNE RIJEČI/KEYWORDS

**Sažetak**

*U cca 100-150 riječi sažeti cijelu radnju. Piše se nakon što je radnja u potpunosti gotova. Pozicionira se na kraju radnje (tako je propisao FESB) ALI GA PROČITA NAJVEĆI BROJ LJUDI (85%) KOJI OSIM NASLOVA UOPĆE NEŠTO OD RADNJE PROČITAJU. ZATO U NJEMU TREBA PRECIZNO SAŽETI ONO ŠTO U RADNJI DONOSIMO NADUGO I NAŠIROKO! ☺☺☺*

**Ključne riječi**

*SQL Server, .NET, Visual Studio, Java, Cloud*

Napomene za pisanje **sažetka** i **ključnih riječi**:

* trebaju ***izgledati TOČNO*** kao gore napisani primjer (osim što sam gore dio teksta istaknuo VELIKIM SLOVIMA)
* sažetak se piše:
  + u trećem licu jednine (bezlično),
  + u pasivu i perfektu (a ne u u futuru i sadašnjosti) – tj. *napravljeno je, analizirano, korišteni su, prikazane su*, itd.
* sažetak nema paragrafa – sve je jedan paragraf.
* na ovoj stranici se pišu **Sažetak** i **Ključne riječi** a na drugoj stranici **Abstract** i **Keywords**
* 5-6 ključnih riječi (mogu biti i složeni pojmovi) koji najbolje odražavaju sadržaj radnje. Nema paragrafa – sve je jedan paragraf. Pojmovi se odvajaju NE TOČKOM I ZAREZOM (;) nego zarezima.

***Naslov na engleskom (prevedi)***

**Abstract**

*Prevedi SAŽETAK na engleski*

**Keywords**

*SQL Server, .NET, Visual Studio, Java, Cloud*